Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

2012

Modeling and simulation for a surf zone robot

Shuey, Eric; Shuey, Mika

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/27905

This publication is a work of the U.S. Government as definedin Title 17, United
States Code, Section 101. As such, it is in thepublic domain, and under the
provisions of Title 17, United StatesCode, Section 105, is not copyrighted in the U.S.

Downloaded from NPS Archive: Calhoun

DUDLEY
KNOX
LIBRARY

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MODELING AND SIMULATION FOR A SURF ZONE ROBOT

by

Eric Shuey
Mika Shuey

December 2012

Thesis Co-Advisors: Richard Harkins
Timothy H. Chung

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704–0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD–MM–YYYY)* 14-12-2012	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* 2012-01-01—2012-12-14

4. TITLE AND SUBTITLE Modeling and Simulation for a Surf Zone Robot	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Eric Shuey, Mika Shuey	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Department of the Navy	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.IRB Protocol Number: N/A

14. ABSTRACT

A kinematic and dynamic model for a three degree-of-freedom surf zone robot is developed and tested with a physical test platform and with a simulated robot in Robot Operating System. Derived from Lagrangian mechanics and relying on angular wheel velocities from encoders, the model successfully demonstrates accurate prediction of motion on simple terrain. The application of the model to future platforms is analyzed and a broad examination of the current state of surf zone robotic systems is provided. An in-depth discussion of the potential improvements to the model is made and the critical work still needed to realize a complete and deployable surf zone platform is described for future study.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
Unclassified	Unclassified	Unclassified	UU	93	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

MODELING AND SIMULATION FOR A SURF ZONE ROBOT

Eric Shuey
Lieutenant, United States Navy
B.S., Systems Engineering, United States Naval Academy, 2006

Mika Shuey
Lieutenant, United States Navy
B.S., Chemistry, United States Naval Academy, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED PHYSICS

from the

NAVAL POSTGRADUATE SCHOOL
December 2012

Authors:	Eric Shuey
	Mika Shuey
Approved by:	Richard Harkins Thesis Co-Advisor
	Timothy H. Chung Thesis Co-Advisor
	Andres Larraza Chair, Department of Applied Physics

THIS PAGE INTENTIONALLY LEFT BLANK

Abstract

A kinematic and dynamic model for a three degree-of-freedom surf zone robot is developed and tested with a physical test platform and with a simulated robot in Robot Operating System. Derived from Lagrangian mechanics and relying on angular wheel velocities from encoders, the model successfully demonstrates accurate prediction of motion on simple terrain. The application of the model to future platforms is analyzed and a broad examination of the current state of surf zone robotic systems is provided. An in-depth discussion of the potential improvements to the model is made and the critical work still needed to realize a complete and deployable surf zone platform is described for future study.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction 1

1.1 Background 1

1.2 Concept of Operations 6

1.3 Research Questions 8

1.4 Scope of Thesis 9

1.5 Methodology 10

1.6 Benefits of Study 11

2 Theory for Mathematical Model 13

2.1 Development of Mathematical Model 13

2.2 Robot Treated as System of Three Bodies 18

2.3 Matlab Implementation 24

3 Simulation and Experimental Setup 29

3.1 Drive Patterns for Testing 29

3.2 Simulation in ROS 30

3.3 Physical Test Platform 35

4 Results 41

4.1 Validation of Model with ROS Simulation 41

4.2 Physical Platform Results 42

4.3 Advanced Testing in Surf Zone Terrain. 49

5 Conclusions and Future Work 65

5.1 Discussion of Final Results 65

5.2 Future Areas for Work 66

Initial Distribution List **75**

List of Figures

Figure 1.1 Bender surf zone platform 2

Figure 1.2 RHex platform from Boston Dynamics 4

Figure 1.3 REMUS 100 AUV from Kongsberg Maritime 4

Figure 1.4 Aqua hybrid platform from McGill University 5

Figure 1.5 MONTe surf zone robot from Naval Postgraduate School (NPS) 6

Figure 1.6 Typical surf zone environment 10

Figure 2.1 Global and robot body coordinate frames 17

Figure 2.2 Diagram of model robot dimensions 20

Figure 2.3 Translation from wheel axis to robot body axis through the parallel axis theorem. 21

Figure 2.4 Diagram of the wheel and robot angular velocity vectors 22

Figure 2.5 Block Diagram for Matlab model program 26

Figure 2.6 Diagram depicting relationship between wheel angular velocity and forward force exerted. 27

Figure 3.1 Sample patterns for test platform. 30

Figure 3.2 Sample ROS architecture 31

Figure 3.3 (a) Profile drawings of Husky (b) Gazebo model of Husky 34

Figure 3.4 Photograph of four wheel prototype platform used for physical testing. 35

Figure 3.5 Photograph of test platform interior construction with servo motor and gear reduction layout. 36

Figure 3.6 Sample of physical platform encoder data text file 37

Figure 3.7 Sample of Matlab code for the conversion of encoder data to wheel angular velocity for the physical platform 37

Figure 3.8 (a) NPS Vicon motion tracking lab (b) Vicon T160 motion capture cameras 38

Figure 3.9 Matlab Code for the conversion of '.bag' file positional data ($bagX$, $bagY$) to velocity ($bagXdot$, $bagYdot$) and forward velocity in the robot frame ($bagxdot$) 39

Figure 3.10 (a)Path in hard pack sand (b) Path diagram from hard sand run 39

Figure 4.1 Forward velocity comparison of ROS simulation and dynamic model results for a straight line run. 42

Figure 4.2 Trajectory comparison of ROS simulation and dynamic model results for a straight line run 43

Figure 4.3 Forward velocity comparison of ROS simulation and dynamic model results for a square pattern 44

Figure 4.4 Trajectory comparison of ROS simulation and dynamic model results for a square pattern 45

Figure 4.5 Comparison between motion capture, kinematic model and dynamic model results for a straight line run. 46

Figure 4.6 Forward robot speed for the straight line run. 47

Figure 4.7 Various parameters as functions of time for the straight line run. 51

Figure 4.8 The robot's X and Y position as determined by the Vicon motion capture setup are presented as a function of time for the square pattern. 52

Figure 4.9 Forward robot speed from motion capture data for square pattern run. . 52

Figure 4.10 The X and Y position and velocity, robot heading, and angular velocity as determined by the Vicon motion capture setup are presented as functions of time for the square pattern. 53

Figure 4.11 Path overview for motion capture and kinematic model for the square pattern 54

Figure 4.12 [Comparison of various results for motion capture and kinematic model for the square pattern . . . 55

Figure 4.13 Forward robot speed from motion capture data and kinematic model results for a square pattern run . . . 56

Figure 4.14 Path overview for motion capture and scaled kinematic model for the square pattern . . . 56

Figure 4.15 Comparison of various results for motion capture and scaled kinematic model for the square pattern . . . 57

Figure 4.16 Comparison of robot trajectory results for a square run. . . . 58

Figure 4.17 Forward robot speed from motion capture, kinematic model, and dynamic model results for a square pattern run . . . 58

Figure 4.18 Various results for scaled kinematic and dynamic models for square pattern . . . 59

Figure 4.19 Vicon results for complex pattern . . . 60

Figure 4.20 Comparison of motion capture, kinematic and dynamic model for complex pattern . . . 60

Figure 4.21 The forward speed of the physical platform as a function of time for the square-spiral-line experiment. . . . 61

Figure 4.22 Various results for scaled kinematic model, dynamic model, and Vicon motion capture for complex pattern . . . 62

Figure 4.23 Overlay pattern of ground truth and dynamic model prediction for hard sand . . . 63

Figure 4.24 Overlay pattern of ground truth and dynamic model prediction for soft sand . . . 63

Figure 5.1 Husky platform in Hector Gazebo . . . 68

Figure 5.2 (a) Test platform with experimental Whegs attached (b) Computer generated design for future Wheg . . . 69

Figure 5.3 Test platform on Del Monte beach using experimental Whegs . . . 70

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

2D	two-dimensional
3D	three-dimensional
CONOPS	concept of operations
COM	center of mass
DOF	degrees of freedom
GPS	Global Positioning System
LCAC	Landing Craft Air Cushion
LCS	Littoral Combat Ship
LIDAR	Light Detection And Ranging
NPS	Naval Postgraduate School
RHIB	rigid hull inflatable boat
ROS	Robot Operating System
UAV	unmanned aerial vehicle
U.S.	United States of America
UUV	unmanned underwater vehicle

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgements

We would like to thank all of our family and friends who supported us at all steps of our time in Monterey. A husband-wife joint thesis was a unique and inspiring experience that we both felt was the only way to go. It brought distinctive demands and surprising challenges and would not have been possible without the amazing time and energy everyone put into helping us.

To our thesis advisors, thank you for your trust, perpetual guidance, and mastery of knowledge. You both were an endless source of motivation and determination to make this project as fulfilling and prodigious as it could be.

To Steve Jacobs, summer interns, and everyone else who helped out, your relentless energy and hands-on effort made this work possible and we thank you.

To our son Jack, who always kept us in check and made sure school never came before family; we thank you for keeping us smiling and reminding us why we work so hard.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

The requirement for robotic systems to operate successfully in a wide variety of physical environments and operational roles has been validated in the past few decades. The ability of autonomous unmanned systems to carry out various missions, with extended on-station time and sensor capability, while removing human risk, has proven extremely valuable. Due to these advantages, a large number of autonomous platforms have been developed and deployed for use on a variety of land-based terrains. Similarly, there have been numerous successful developments of autonomous aquatic vehicles that are capable of operating in shallow water. The ability for a vehicle to transition between these environments expands the realm of operational application. It benefits from ease of access from the sea while gaining on-ground proximity and higher sensor fidelity compared to aerial and remote sensors. Projected mission capabilities are robust such as surveillance, environmental monitoring, intelligence, reconnaissance, and mine clearing.

With this capability comes significant difficulties. Operations in the surf zone and the transition from sea to shore introduce a variety of environmental and dynamic factors. These factors are extensive and dramatically differentiate the design criteria for unmanned systems exclusive to sea or land.

Over the past few years, a collaboration between the NPS and Case Western Reserve University, sponsored by Temasek Defence Systems Institute, has sought to develop a platform to exploit this littoral region. The importance of this region is well understood for both military and civilian applications. Several prototype mechanical platforms have been developed using biologically inspired locomotion to traverse the surf zone. Limited work has been done on the simulation and modelling of such platforms in the environment and is critical to the ultimate capabilities and success of an autonomous surf zone robot.

1.1 Background

1.1.1 Idea of Biologically Inspired Surf Zone Robot

Motivation for platform concepts has stemmed largely by observing organisms capable of locomotion and survival in difficult terrains. By modeling these biological features, a history of platforms has been produced that mimic the biological locomotion and show great promise in a suitable system for employment in the surf zone. While biological motion is not an absolute requirement for success, it provides an immediate and proven source of inspiration, particularly in a new and emerging field such as surf zone operations.

1.1.2 Methods of Locomotion in Non-trivial Environments

Initial work at NPS on a surf zone vehicle focused on negative buoyancy using tracked platforms that used traditional tank drive mechanisms to remain on the sea floor through the surf zone transit and drive on to the beach.

Provided by the Surf Zone Crawler Group of Naval Surface Warfare Center Panama City, a Foster Miller Lemming tracked platform, as shown in Figure 1.1, was fitted with basic logic and control hardware. While not waterproof, it demonstrated the effectiveness of differentially driven vehicles in the sandy beach environment. Further modifications allowed for remote control basic and way point navigation. This platform served as the initial test bed for future surf zone autonomous platforms at NPS [1].

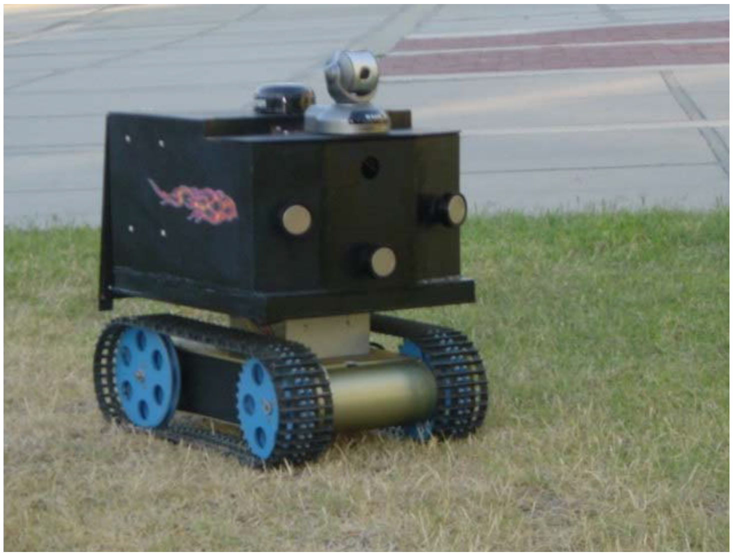


Figure 1.1: NPS robotic platform Bender comprised of a Foster Lemming tracked base and watertight housing for onboard sensors and processing. From [1]

Researchers at Case Western in 2001 began to develop a platform to mimic the locomotion of a cockroach. The Wheg$^{\text{TM}}$ is a simple combination of the traditional motor driven efficiency of a wheel with the traction and obstacle scaling of a leg [2]. Further work elaborated this concept into an entire drivetrain concept. Focused on the simplicity and efficiency of components, the drive configuration employed a six Wheg™ platform with a single drive motor supplying torque to all the Whegs™ through an internal chain drive. The gait was passively set to an alternating tripodal stance with a compliant axial spring in the hub of the Whegs™ to allow the phase to align laterally to assist in climbing larger obstacles. Ackerman steering [3] was employed on the front Whegs™ for directional control of steering. The overall setup was successful in climbing relatively large obstacles with minimal equipment and power requirements [4].

To further advance the platform and better simulate the dynamic locomotion of the cockroach, later work incorporated a hinged body about the center Whegs™. This hinge allowed the platform to actively raise its center of gravity, putting larger torque and mass over the front Whegs™ in contact on top of an obstacle. This modification added significant obstacle scaling and clearing of the center structure from high centred obstructions. With proven performance gains, but significantly more complex internal engineering, the concept was further simplified with the move to a four Wheg™ differentially skid steered drive configuration. In lieu of a hinged body, an active tail was developed that could be stowed and deployed as needed to accomplish the same performance gains as the hinged body without having to pass internal chain drives and engineering requirements of a split chassis. The final iterations of Case Western's work demonstrated that a simple and robust platform could be modeled after biologically inspired locomotion on surface terrain [5].

Of great importance in the surf zone is the durability and ruggedness of the platform and its method of locomotion. The harsh and dynamic environment combined with the variety of different terrain demands a simple and reliable configuration. In addition, it must be easily waterproofed and capable of carrying the necessary sensors. An example of such a platform is RHex from Boston Dynamics, as shown in Figure 1.2. Using six independent curved legs and rugged chassis with in-body sensor windows, it is capable of scaling formidable terrain in wet and dry environments as well as surviving large impacts. The curved legs give it some ability to paddle on the surface of water and its vertically symmetrical layout improves mobility and operation independent of its orientation [6] .

1.1.3 Methods of locomotion in water and hybrid platforms

Autonomous systems dedicated to the water-borne environment are usually designed to minimize the excessive drag associated with motion in a fluid environment. The REMUS system has been in use by the United States of America (U.S.) Navy for several years. It employs a modular sensor suite for a wide variety of mission capabilities such as hydrographic survey, mine

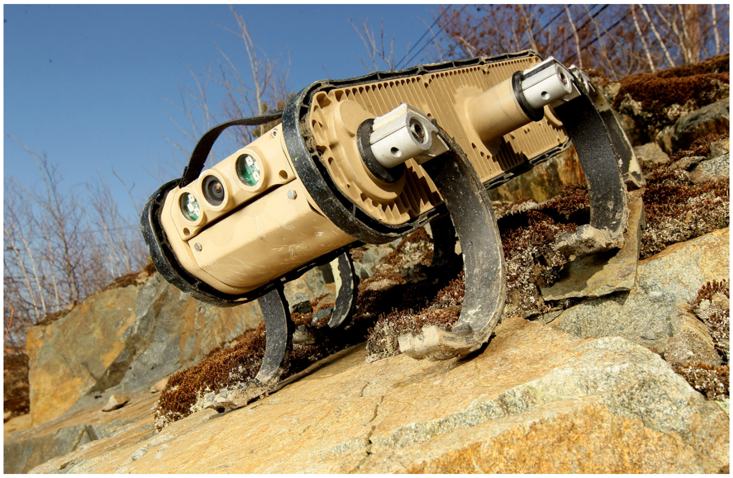


Figure 1.2: The RHex platform from Boston Dynamics incorporating independent drive legs, integrated sensors and durable waterproof body capable of scaling very challenging terrain. From [6]

detection, environmental monitoring and data sampling. The REMUS 100 and 600, as shown in Figure 1.3, closely resemble torpedo profiles with minimal protrusion for the necessary sensors and control surfaces [7]. Bridging the mobility gap between sea and land is challenging because of the different forces and requirements on the platform. Torpedo shapes do not interact well with physical terrain and are unsuited for locomotion on land. In an attempt to compromise, hybrid type platforms have been developed to blend the hydrodynamic shape of a torpedo with the terrain scaling capability of wheeled and legged platforms. Aqua, as shown in Figure 1.4, is an amphibious platform developed by McGill University that utilizes six independent legs that are capable of paddling in water similar to a pinniped. Providing slow, smooth six DOF motion in water, the vehicle is also capable of locomotion on land using an alternating tripodal gait. While one of the only platforms capable of this transition, its efficiency in one realm must be sacrificed for the other and only marginal efficiency is capable on a single set of legs for both water and land [8].

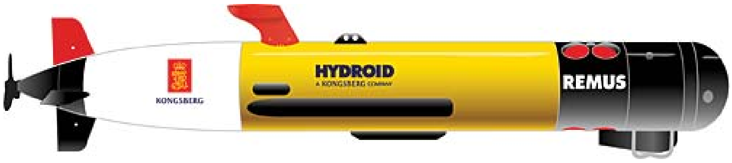


Figure 1.3: The REMUS 100 AUV platform from Kongsberg Maritime capable of dedicated shallow water operations to depths of 100 meters with a variety of onboard sensors. From [7]

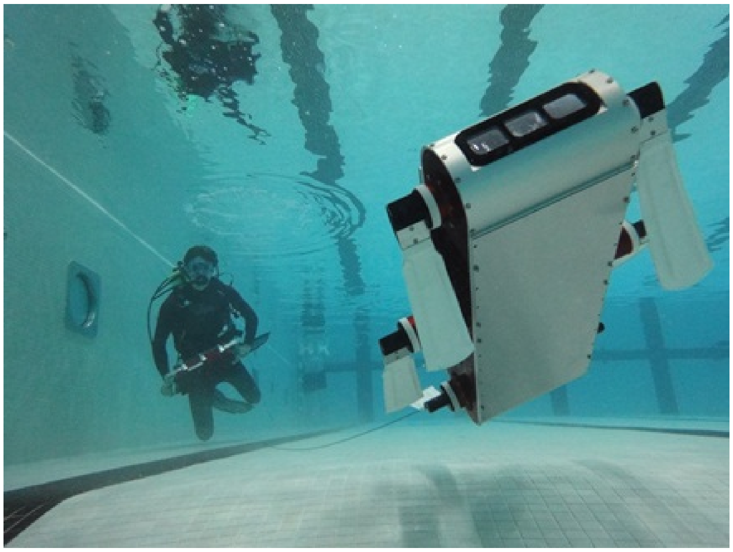


Figure 1.4: Aqua hybrid platform from McGill is capable of effective waterborne propulsion and beachside locomotion using paddles similar to a marine mammal. From [8]

1.1.4 NPS Surf Zone Robot Project

Leveraging these examples and collaborations, NPS developed several platforms around the concept of surf zone operations. Initially, systems focusing on single design accomplishments with further evolutions resulting in more complex and capable designs tailored for the many demands of operation in the surf zone.

Originally built by Case Western, Agbot was a Wheg™ based system utilizing Ackerman steering, single drive propulsion, six passive torque compliant drive shafts and no suspension. NPS added a waypoint navigation system with a Rabbit processor that was programmed in Dynamic C. Successful field testing was done in the beach environment but the platform was not suitable for true surf zone operations due to a lack of waterproofing [9]. Further designs were demonstrated on the Robster platform which was the first attempt at automating the use of a tail [10].

The most recent platform was a culmination of the lessons learned from prior designs as well as significant work towards a rugged and waterproof chasis. Built on a standard Pelican case, MONTe, as shown in Figure 1.5, was a four Wheg™ system with two drive motors for differential steering. Utilizing in-house three-dimensional (3D) polycarbonate printing, an extensive suspension and tail system were designed. This was also the first excursion into the open source Robot Operating System (ROS) and advanced on-board sensing including Global Positioning

System (GPS), solid state accelerometers, and gyroscopes [11]. The overall system proved capable of travel over adverse terrain including stairs, sand, and slopes. The additional sensors allowed for the implementation of the autonomous tail and GPS for simple waypoint navigation. Waterproofing was accomplished through gaskets and seals with minimal success [12].

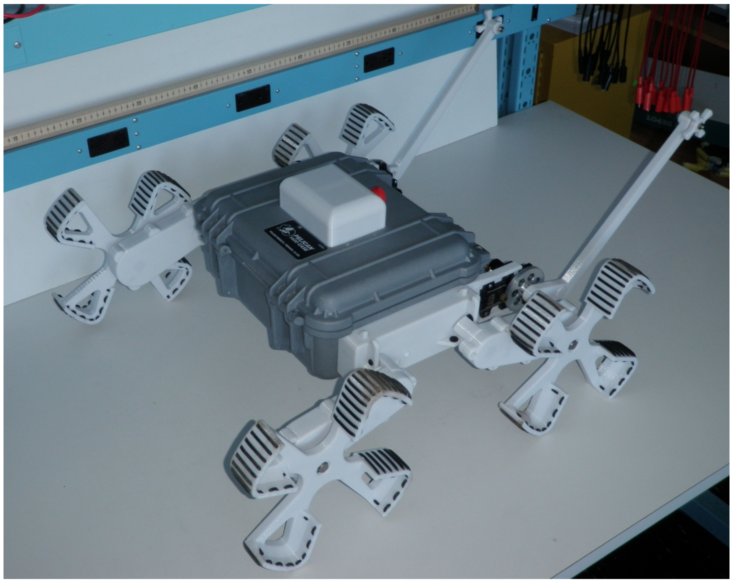


Figure 1.5: MONTe surf zone robot utilizing a commercial waterproof polycarbonate case and 3D printing for suspension and Wheg™ construction. From [12]

1.2 Concept of Operations

The background of systems discussed are influential in providing initial implementation of the various subsystems and lessons learned from their testing in the surf zone terrain. When combined with the scope of research in this field, an overall design and ultimate concept of operations (CONOPS) is developed. The driving consideration in our platform is a single man-portable, low cost and scalable system that can be employed in large numbers to adequately accomplish mission goals. Due to the harsh nature and difficulty of operation in the surf zone, it is expected that some vehicles may become incapacitated or lost due to damage, terrain, and/or system failure. Similar to swarm operations, the loss of a single robot is not mission-critical and mitigated by adjacent units.

Being a small and light weight platform, the physical deployment is simple and versatile, such as hand placing the vehicle in the water. The ease and flexibility of launch lends to a large number of systems being launched from a variety of platforms over a short period of time.

Deployment could also be done by sea, air or land. Possible launch platforms include small craft such rigid hull inflatable boat (RHIB) and rafts, aerial vehicles such as helicopter drops from low height, and larger craft capable of delivering significant numbers of vehicles such as traditional surface ship combatants, amphibious well decks or Littoral Combat Ship (LCS). An additional means of covert deployment, when necessary, would be through submarine and unmanned underwater vehicle (UUV)s. Deployment distance would be flexible and dictated by the specific capability of the launching vessel and the necessary time frame for it to reach the operating station. Distances from as close as 100 meters from shore by small craft to as far as a few nautical miles by larger vessels would be possible and dictated by the endurance and on-station time of the individual robot. Another possibility for maximizing on-station time would be to directly place robots on the shore via helicopter or Landing Craft Air Cushion (LCAC), removing the initial water-borne travel, but still allowing for the dispersive return of the platform by sea after its mission.

While traversing from sea to land, individual platforms rely on a primary surface propulsion mode of jet drive or hybrid wheel paddling. The vehicle maintains slight positive buoyancy but also be capable of shallow submersion to a depth of one to two meters for covertness or surface obstacle avoidance. GPS waypoint guidance and on-board sensors such as low power sonar or camera vision are capable of detection and avoidance of water-borne obstacles such as kelp, shallow reefs, and man made obstructions. This water propulsion continues through the surf zone. Due to wave action or sea state, the platform is operable in either upright or inverted positions. Once shallow water is reached to allow traction with drive wheels, locomotion transitions the platform into terrain based mode.

Once onshore, the vehicle uses on-board sensors such as Light Detection And Ranging (LIDAR) and GPS to navigate the beach environment. Unlike unmanned aerial vehicles (UAV), which require constant power to remain airborne, the on-station time and minimal idle power while loitering for ground vehicles is substantially longer and provides much closer and detailed persistent analysis of the area. Because of these advantages, a wide range of mission capabilities exist such as:

- Intelligence gathering
- Location or beacon marking
- Submerged and surface mine and obstacle detection
- Data gathering via acoustic, RF, visual sensors

- Communications relay for shore to sea link
- Obstacle or mine clearance via suicide explosive

Since the vehicle is located in both terrestrial and water-borne environments, it can utilize the full gamut of communications capabilities. The standard assortment of RF systems including UHF and satellite radios can be outfitted, as well as acoustic modem systems such as Sea-Web [13]. Not being severely limited by weight restrictions compared to a UAV, a surf zone vehicle can drive on station then position itself for physically static communications and sensor relay. This unique role for ground vehicles uses a minimal amount of energy while maintaining significant capabilities.

Due to the inexpensive and larger number of vehicles being deployed, the complete recovery of all assets is an optional requirement and dictated by the specific mission or opportunistic discoveries during a mission. For example, if hidden mines or obstructions was to be discovered in the planned path for a traditional amphibious attack, the required number of vehicles would be positioned in close proximity to these targets. They then could loiter on these targets until on-board explosives could be detonated simultaneously among all suicide designated vehicles just moments before the amphibious landing were to commence. This could provide a stealth and minimal risk beach clearance while only sacrificing individual inexpensive robots for the removal of specific hazards.

Remaining vehicles return to sea utilizing the combination of jet drive and wheeled locomotion. The same sensor suite and maritime object avoidance utilized for the initial sea to shore transit is used for the return transit. Individual vehicles are recovered via small craft, submerged well deck vessels, or specialized recovery equipment from traditional ships.

The combination of small, inexpensive, and capable vehicles provides a significant and varied mission profile with on-station time beyond the traditional unmanned aerial systems. Unique capabilities such as the suicide mine clearance and static communications relay add great value to combatant commanders and vastly improve our ability to utilize and exploit the surf zone region with unmanned systems.

1.3 Research Questions

As noted in the various examples in Section 1.1, the physical configuration and construction of surf zone robots has seen much attention and demonstrated success in vehicles that can physically maneuver and survive in this environment. Limited work has been performed in the simulation and modeling analysis of such vehicles and is where our work focuses. Some of the primary research questions pertaining to this research area include:

- How can a robotic platform reliably operate in a surf zone environment?
    - How can a three degree of freedom kinematic and dynamic model of a differentially driven vehicle be used to provide useful and predictive data for design and implementation of a prototype platform operating in the surf zone environment?
    - How can a physically simulated platform provide comparable data for predictive model testing and be scalable for more complex simulation in the expected terrain and obstacles in the surf zone environment?

These questions are fundamental to the final configuration of a successful and reliable surf zone platform. By answering these questions, ground work will be laid out for the few remaining challenges of incorporating the successful physical platforms with the necessary autonomy into a complete and deployable system.

1.4 Scope of Thesis

The scope of research is limited to the beach, defined as the waterline and inland for a distance of 100 meters with operation in the waterborne environment excluded from research. An example of a typical surf zone enviorment is shown in Figure 1.6. Points of investigation include development of mathematical dynamic model to predict and analyze the motion of a differentially steered vehicle. Initial models assume a two-wheeled differentially steered vehicle with no slip. Later models assume a four-wheel differentially skid steered vehicle with no slip in line with traction and allowable transverse slip. The inertia of the vehicle body and wheels is accounted for in the models. The models for waterborne stability and specific interaction between the various terrain features and vehicle locomotion is beyond the scope of this work. This initial analyze is of a smaller DOF system and serve as the groundwork for potential future work into the more complex interactions of specific features and water borne motion. Additional investigation is performed into simulating a platform for data production as well as creation of more complex and accurate terrain to mimic the surf zone environment. Typical obstacles might include slopes, valleys, washouts, rocks, cliffs and man-made obstructions. A simulated environment is constructed to represent the beach zone and these relevant obstacles as well as to test the capabilities and success of the navigation and control logic.

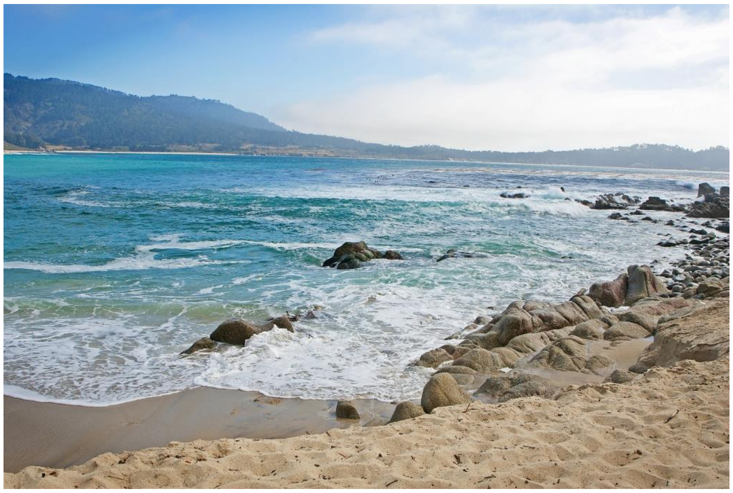


Figure 1.6: A picture Carmel River State Beach near Monterey, California showing a typical surf zone enviroment. Features include submerged and exposed rocks, hard and soft sand, swell and breaking surf.

1.5 Methodology

Using Lagrangian mechanics, a three degree of freedom mathematical model is developed from the kinetic and potential energy of differentially steered robot. These dynamical equations are implemented in Matlab for visualization and assessment of the models ability to accurately predict motion based on wheel angular velocities. The results from the Matlab model are then be compared to real world platforms utilizing 3D motion tracking and ROS simulation.

Simulation is developed and coded in the open source Robot Operating System (ROS) running on the Linux based Ubuntu operating system. ROS is one of several academic standards for robotic system control and benefits from a large and active community [14]. A critical capability that ROS provides is integration with Gazebo, a 3D visual simulation environment where code designed for physical robots, sensors and manipulators are tested with realistic physics [15]. In Gazebo, an established model platform, Clearpath Robotics Husky A200, is used as the test bed [16]. The pace of iteration and design is significantly increased in the simulated world compared to physical testing. Redesigns and full system modifications can be performed in minutes vice the longer time scale to physically test, reprogram and return to the surf zone. In addition, Gazebo is tailored to create a realistic surf zone environment complete with elevation change, friction coefficients, standalone obstacles, and dynamic movement. Code written and tested in ROS with Gazebo is implementable on a physical robot with little if any alteration or modification. This software suite allows for the complete development and testing of the complex environments for a surf zone robot as well as the eventual implementation in future physical platforms.

1.6 Benefits of Study

The littorals have become an increasingly vital and exploited region due to the universal need to transition from sea to shore [17]. This transition of operating in the littorals also presents great challenges due to the wide range of surf, sand, rock, and marine plant life that reside in the region. Despite this established importance of the littorals, no commercial or deployable option for a surf zone robot was found in the scope of our research. The increased capability of having a covert, mobile, and embedded sensor suite on the ground capable of traversing the varied and hostile terrain would provide a much wider range of reliable detection and intelligence against such natural and integrated defences as covert mines, underwater obstacles, and obscured terrain. In addition, the communications and sensors relay capabilities afforded by a large number of distributed small and inexpensive vehicles in the water and on the beach would provide an immense tool for mission planning and command and control of the battle space. Communities ranging from Naval Special Warfare to environmental research could benefit from the many capabilities this specialized system could provide.

A limited but focused amount of research completed on surf zone robotic systems and several preliminary designs have shown promise but little work has been done on the control logic, autonomy, sensors and required modeling of such systems [5], [9], [11], [12]. Our work strives to address a remaining hurdle of accurate modeling and simulation to complete the robotic system and allow the full optimization and integration of such a platform in future work.

This thesis is organized as follows. A background and history of platforms with relevant contributions to operations in the surf zone is discussed. A mathematical model from the Lagrangian equations of motion is developed and tailored for the specific locomotion expected in the surf zone. In order to test this model, a variety of physical and simulated runs are developed. Motion tracking and data logging are processed and the resulting data is compared to the predictions of the model and analysed for accuracy and points of failure. Finally, a summary of the conclusions drawn is made and the future work necessary to integrate this work and other critical features discovered during research are discussed for the successful development of a surf zone robot.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Theory for Mathematical Model

This chapter describes the process used to develop both a kinematic and dynamic model of a differentially driven vehicle. Section 2.1 explores the design requirements of a model, techniques such as Lagrangian Mechanics to derive mathematical equations of motion, and also chosen assumptions and simplifications. Section 2.2 describes the derivation of the equations of motion specific to a particular four wheel test platform. Finally, Section 2.3 describes the steps required to produce trajectory and velocity results in Matlab from the derived mathematical model.

The following steps are used in the development of a mathematical model and are described in greater detail in Sections 2.1 and 2.2.

1. Determine purpose of model.
2. Determine mathematical techniques that will be employed.
3. List assumptions and simplifications.
4. Formulate justification for a three degrees of freedom (DOF) model.
5. Determine the kinetic energy of each component of the robotic system.
6. Determine the Lagrangian, L.
7. Use the Euler-Lagrange equation to formulate a set of dynamic equations.
8. Integrate the dynamic equations twice to develop the equations of motion.
9. Determine the initial conditions such as position,velocity, mass, and dimensions of the system.
10. Determine the inertia of each component of the system.
11. Determine a relationship between binary signals from a wheel encoder to angular velocity of the robot's wheels.
12. Develop a relationship between the forces and torque of the robot ($F_x, F_y, F_L, F_R, and \tau$), angular wheel velocity.

2.1 Development of Mathematical Model

This section provides an introduction to the process used to derive a predictive model for motion of a robotic platform. First, the requirements of the motion model are identified. Next, a variety of techniques to derive a model are explored and assessed for applicability to a surf zone robot. Finally, assumptions and constraints are imposed in order to reduce the complexity of the mathematical derivation.

2.1.1 Requirements of Model

The intended purpose of the model developed in this chapter is to provide accurate and predictive estimates for the motion of a four-wheeled skid-steered mobile robot operating in a surf-zone environment. The predictive motion estimates are then to be used for performance analysis of vehicles on various terrain types, mobility improvement in closed loop control, and position determination independent of any external sensors such as GPS. Because the NPS Surf Zone project is ongoing and the platform is constantly being adapted, the model is designed in a way that specific parameters such as mass and dimensions can be changed based on the robot design. Most NPS Surf Zone robots have utilized skid steering as the preferred steering method because it provides strong traction and high mobility. Therefore, the model should be able to simulate this steering method.

An important design criterion of the model that is considered is the ability to account for the challenges presented in a complex beach environment. The effects of variables such as surface consistency including sand, gravel, and rocky beach terrain, varying slope and elevation, and hydrodynamic forces from surging swells are all criteria that are considered. A foundational model should have the flexibility to accommodate these dynamic variables in order to provide an accurate trajectory.

Furthermore, any data gathered for the model need to come from simple, reliable, and low-cost equipment. The survivability of the platform in a rugged surf zone environment depends on simplicity and durability of the components. Therefore, it is preferable that the model be designed to require inputs from basic sensors that are typically already found on past NPS Surf Zone robots.

2.1.2 Techniques to Derive Model

One of the main challenges in accurately predicting the motion of a skid-steered robot is the difficulties presented in modeling the wheel/ground interactions of the tires as well as the kinematic constraints of the platform. A four-wheel, skid-steered robot requires skid in order to change heading but this skid is not easy to predict or measure. A kinematic model can be derived for the purpose of position estimation and orientation based on a robot's parameters, but actual motion estimation shortcomings caused by the presence of wheel slip affects the accuracy of the model [3]. There are several approaches to the development of a kinematic model that can either account for these difficulties or bypass them altogether. Attempts have been made by previous researchers to model skid-steered mobile robots by combining inertial measurements with a wheel estimation schemes using the kinematic relationship between wheel slip and instantaneous centers of rotation of the wheels [18], [19]. These techniques may provide accurate localization but are complicated and require sensors that are not readily available on our platform. Other researchers have created methods for localization by combining odometry-based position estimation with visual landmarks [20]. A limitation of this work is that although it may provide accurate position estimation, it requires that the robot operate in a known or well-structured environment. Furthermore, these kinematic models may not be easily adapted to account for the effect of external forces on the robotic platform such as those exerted by waves and other water phenomena.

Another approach for localization is through the development of a dynamic model. A dynamic model provides two relevant and useful relationships, namely the force-mass-acceleration and torque-inertia-angular-acceleration equations [21]. These relationships show us what force and torque inputs are required to exert a desired acceleration and angular acceleration of a platform. This type of information can be applied to several different types of motion planning, including 1) time optimal motion planning, 2) energy efficient motion planning, 3) reduction in the frequency of replanning, and 4) planning in the presence of a fault, such as a flat tire or faulty motor [22]. A dynamic model is particularly useful in the case of a surf-zone robot because it would be able to account for forces exerted by waves and forces associated with the type of terrain being traversed. This type of dynamic model is our goal, however, the model presented in this thesis is a foundational model that leverages simplifying assumptions to support an initial development as described later in this chapter.

2.1.3 The Beginning of a Dynamic Model: A Review of Lagrangian Mechanics

Lagrangian mechanics provide a useful method to derive a force based equation of motion set based on a conservation of energy within a system.

The **Lagrangian** goes as $L = T - U$ where T and U are the system's kinetic and potential energy, respectively.

The Lagrangian is then used in the Euler-Lagrange equation,

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = 0, k = 1, ..., N \tag{2.1}$$

where q and $\dot{q}$ are the generalized coordinates and velocity, respectively. These equations are integrated twice to find the equations of motion as done in Section 2.2.4. In this initial model, we assume that the robot is operating on a flat surface and therefore the gravitational potential energy is constant.

The kinetic energy can be found by the summation of the kinetic energy of each rigid body in the robot system:

$$T_{total} = \sum_{i=1}^{n} T_i. \tag{2.2}$$

The kinetic energy of each component, in turn, is the sum of the kinetic energies due to translation and rotation:

$$T_i = \frac{1}{2} m_i {v_i}^2 + \frac{1}{2} I_i \omega_i^2, \tag{2.3}$$

where m_i is the mass, $\vec{v}_i$ is its velocity, ω_i is the angular velocity, and I_i is the inertia about the center of mass (COM) of I. Further, the inertia tensor,I, can be written in matrix form as:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}. \tag{2.4}$$

For a body with continuously distributed mass, each individual moment of inertia within the inertia tensor can be found with the following equation: $I = \int r^2 dm$ [23]. By taking the derivative of $\rho = \frac{m}{V}$, where V is the volume, dm can be found in rectangular coordinates to be $dm = \rho dV = \rho dxdydz$, where ρ is the density of the body.

There are sophisticated ways to determine the exact inertia of a platform, but this initial model assumes that the platform is cubic, with uniform density and a center of mass at the center point of each component. Similarly, the wheels are treated as cylinders. These and other assumptions are detailed below.

2.1.4 Assumptions and Constraints

The following are assumptions and simplifications are leveraged in the implementation of the model:

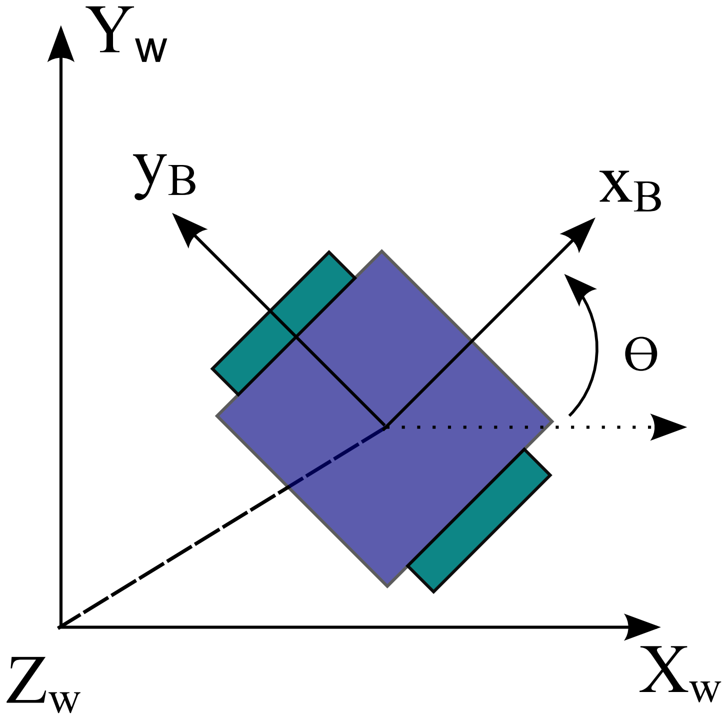


Figure 2.1: The global coordinate frame and robot coordinate frame used for the model derivation, denoted by uppercase with subscript W and lowercase with subscript B, respectively.

1. **The robot is modeled as two wheeled differentially steered rather than four wheeled skid-steered.** Skid-steering has been shown to be more effective than differential steering in difficult outdoor environments, particular in difficult terrains where strong traction and high mobility are required [22]. Past NPS surf zone robotic group physical platforms have employed this steering technique with four and six wheels. The model presented here in only considers a two-wheeled differential steered robot rather than a skid-steered version to reduce the complexity of the problem. By assuming a differential steer mechanism, we assume the concept of rolling contact with zero lateral slippage. This adaptation reduces the complexity of the model and therefore the accuracy, but it closely approximates the four-wheeled robot with a skid-steering slip coefficient.
2. **The model uses the global and local reference frames shown in Figure 2.1, and has three degrees of freedom: two for position in lateral and longitudinal directions (**x **and** y**), and one for orientation about the vertical** z **axis.** The robot itself is modeled

with three components: the body chassis and two fixed standard wheels. The body frame is defined such that the y axis intersects the center of each wheel. Imposing this simplification means that the movement of internal parts is not taken into account.

3. **Motion is restricted to the horizontal plane and therefore will not account for any changes in elevation.** The inclusion of only 2-D planar motion reduces the number of degrees of freedom from six to three, that is, lateral and longitudinal motion and heading of the robot. Dynamics in the z axis can be added in at a later time if desired. In water, our physical platform is intended to be passively stable and therefore pitch and roll are initially ignored. Similarly, if the physical platform is positively buoyant then the platform always returns to the surface and changes in vertical height are therefore not of concern. Later iterations of the model can be adapted to reflect changes in elevation and depth with minimal difficulty.

2.2 Robot Treated as System of Three Bodies

As shown in Figure 2.1, capital letters with subscript W have been used to denote the universal reference frame while lower case letters with subscript B describing the body frame. The origin of the body coordinate frame is defined to be at the center point of the robot, which is assumed to be the robot's center of mass. The y axis passes through the center of both wheels.

Variables used in the development of this model are listed in Table 2.1. Note that subscript B, L, and R indicate body, left wheel, and right wheel components.

The total kinetic energy can be written as the sum of the kinetic energies of each component, $T_{total} = T_B + T_{wR} + T_{wL}$. For the main body, B, kinetic energy can be written as

$$T = \frac{1}{2} m_B \vec{v}_B^2 + \frac{1}{2} I_B \dot{\theta}^2 \tag{2.5}$$

If velocity is separated into its x and y components then $\vec{v}_B = [\dot{x_B}, \dot{y_B}]^T$ and the total kinetic energy of the body can be rewritten as

$$T = \frac{1}{2} m_B \dot{\vec{x}}_B^2 + \frac{1}{2} m_B \dot{\vec{y}}_B^2 + \frac{1}{2} I_B \dot{\theta}^2 \tag{2.6}$$

Table 2.1: Mathematical Model Variables

Symbol	Variable	Units
v_B	velocity	$\frac{m}{s}$
m	mass	kg
ω_B, $\dot{\phi}_L$, $\dot{\phi}_R$	Angular velocity	$\frac{rad}{s}$
I	Inertia	$kg \cdot m^2$
ρ	Density	$\frac{kg}{m^3}$
r	wheel radius	m
a	chassis length	m
b	chassis width	m
c	chassis height	m
d	axle length	m
h	perpendicular distance	m

2.2.1 Determining the Inertia of the Body, I_B

By defining the origin to be at the center of mass, the products of inertia is equal to zero and the inertia matrix is written as:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \tag{2.7}$$

The inertia about the x axis, I_{xx}, can be written as:

$$I_{xx} = \int (y^2 + z^2) dm. \tag{2.8}$$

Next, dm can be replaced by taking the derivative of density, $\rho = \frac{m}{V}$, such that

$$dm = \rho dV = \rho (dxdydz). \tag{2.9}$$

I_{xx} is now rewritten as:

$$I_{xx} = \rho \int_{-x/2}^{x/2} \int_{-y/2}^{y/2} \int_{-z/2}^{z/2} y^2 + z^2 \, dx \, dy \, dz. \tag{2.10}$$

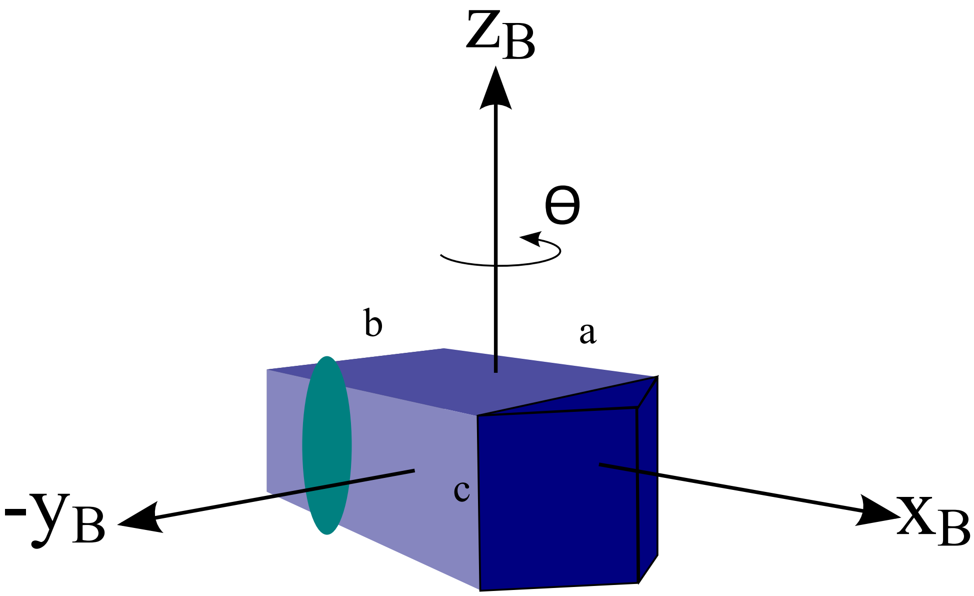


Figure 2.2: *a*, *b*, and *c* are defined to be scalar values for the length, width, and height of the robot, respectively.

Note that x is defined to be the body length along the x axis. As shown in Figure 2.2, the variables a, b and c are now defined to be equal to the robot's length, width, and height, respectively. Using these variables and defining the total mass of robot system as M, where $M = m_b + m_w$, Equation 2.10 can be simplified to the following:

$$I_{xx} = \frac{M}{12}(b^2 + c^2). \tag{2.11}$$

Similarly, I_{yy} and I_{zz} can be written as

$$I_{yy} = I_{bb} = \frac{M}{12}(a^2 + c^2) \tag{2.12}$$

and

$$I_{zz} = \frac{M}{12}(a^2 + b^2). \tag{2.13}$$

Thus, the inertia tensor can be rewritten as

$$I = \frac{M}{12}\begin{bmatrix} b^2 + c^2 & 0 & 0 \\ 0 & a^2 + c^2 & 0 \\ 0 & 0 & a^2 + b^2 \end{bmatrix}. \tag{2.14}$$

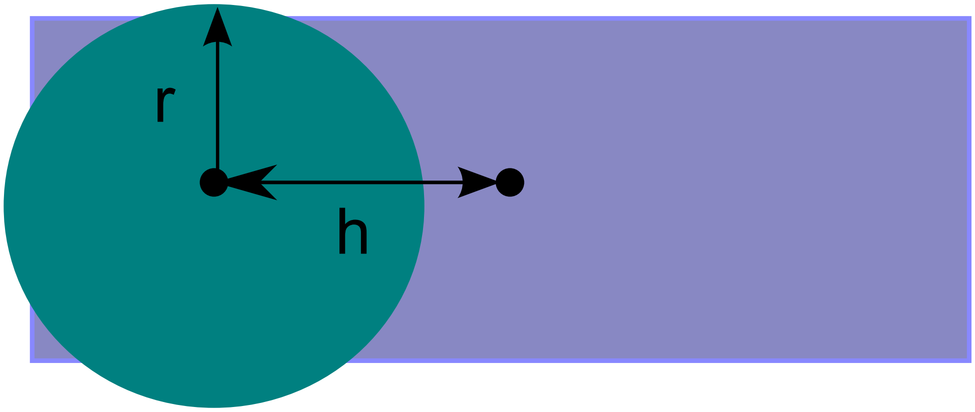


Figure 2.3: Translation from wheel axis to robot body axis through parallel axis theorem. Note that h is the distance between the two axes.

For the purposes of the dynamic model derivation, the moment of inertia along the y axis is the only inertia term of interest. Therefore, I_{yy} is defined to be I_R, the inertia of the robot, and this term is used later to determine the rotational kinetic energy.

2.2.2 Inertia of the Wheels

If each wheel is treated as a thin solid cylinder then the moment of inertia can be approximated to be $I_w = m_w r^2$. However, the wheel moment of inertia is about the wheel's COM and must be translated to the axis intersecting the robot's COM. This can be done using the parallel axis theorem, $I_{w,new} = I_w + m_w h^2$, where h is the perpendicular distance between the wheel's axis of rotation and the robot's axis as shown in Figure 2.3 [24].

2.2.3 Kinetic Energy of the Wheels

As shown in Figure 2.4, the wheel angular velocities are defined as $\dot{\phi}_{wL}$ and $\dot{\phi}_{wR}$. The wheels are offset from the robot's y axis by a distance h as described in the previous section, and are rotating perpendicular to this axis. The total kinetic energy of each wheel are written as the sum of the translational and rotational kinetic energy [24]. The following are the kinetic energy equations for the left and right wheel:

$$T_{wL} = \frac{1}{2} m_{wL} v_B^2 + I_{wL} \dot{\phi}_{wL}^2 \tag{2.15}$$

$$T_{wR} = \frac{1}{2} m_{wR} v_B^2 + I_{wR} \dot{\phi}_{wR}^2. \tag{2.16}$$

These expressions are relevant to the formulation of Lagrange's Equations of Motion, described in the next section.

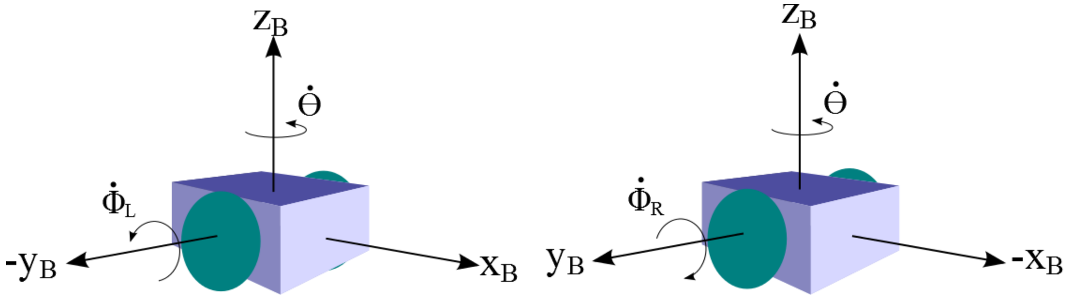


Figure 2.4: Depiction of the wheel angular velocities, $\dot{\phi}_L$ and $\dot{\phi}_R$, and the robot's angular velocity, $\dot{\theta}$.

2.2.4 Lagrange's Equations of Motion

Combining the kinetic and potential energy terms given by Equations 2.6, 2.15, and 2.16 results in the Lagrangian, L:

$$L = T_B + T_{wL} + T_{wR} - U, \tag{2.17}$$

$$L = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}m\dot{y}^2 + \frac{1}{2}I_R\dot{\theta}^2 + \frac{1}{2}I_{wL}\dot{\phi}_w L^2 + \frac{1}{2}I_{wR}\dot{\phi}_w R^2 - U(x, y, \theta, \phi_w L, \phi_w R). \tag{2.18}$$

Note that $m = 2m_w + m_b$ is the total mass and that we have opted to not include the rotational kinetic energy because we have chosen to used the generalized coordinates for longitudinal axis, lateral axis, and heading, $q = (x, y, \theta)$. We have also made the assumption that the robot has zero potential energy. With these assumptions, the Lagrangian simplifies to $L = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}m\dot{y}^2 + \frac{1}{2}I_R\dot{\theta}^2$, and the Euler-Lagrange equation yields the following:

$$\frac{d}{dt}\begin{bmatrix} m\dot{x} \\ m\dot{y} \\ I_R\dot{\theta} \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ \tau \end{bmatrix}, \tag{2.19}$$

where F_x and F_y are the forces exerted by the robot in direction of the x and y axes, respectively, and τ is the torque of the robot. F_x, F_y and τ are all functions of time and related to the wheel angular velocity as discussed in Section 2.3.1. These dynamical equations of motion can be integrated twice to find the following:

$$x(t) = \frac{1}{2}\frac{F_x(t)}{m}t^2 + \dot{x_0}t + x_0 \tag{2.20}$$

$$y(t) = \frac{1}{2}\frac{F_y(t)}{m}t^2 + \dot{y_0}t + y_0 \tag{2.21}$$

$$\theta(t) = \frac{1}{2}\frac{\tau(t)}{I_R}t^2 + \dot{\theta}_0 t + \theta_0. \tag{2.22}$$

2.2.5 Dead Reckoning Kinematic Comparison

Although a dynamic model would be better adaptable to a surf-zone environment, we choose to also develop and employ a kinematic model for a baseline comparison. This model is two-wheel differentially steered with zero slip, and uses dead reckoning based on the angular velocity of the left and right wheels for position estimation.

The following equation relates the wheel parameters with forward velocity of the robot in the global reference frame:

$$\dot{x} = r\frac{\dot{\phi}_L + \dot{\phi}_R}{2}, \tag{2.23}$$

where r is the wheel radius and $\dot{\phi}_L$ and $\dot{\phi}_R$ are the angular velocity of the left and right wheels as defined in Section 2.2.3. The number of wheel revolutions completed in one time step is calculated from the integration of the wheel angle,ϕ_L or ϕ_R over the time interval of interest. For the left and right wheels, this quantity is defined as REV_L and REV_R, respectively. The change in the number of revolutions is next defined as ΔREV_L and ΔREV_R. Additionally, d is defined as the axle length and therefore the robot's change in heading in one time step is written as:

$$\Delta\theta = 2\pi r\frac{\Delta REV_L - \Delta REV_R}{d}. \tag{2.24}$$

The change in heading, $\Delta\theta$ is summed to compute the heading at each time step, and the X_W and Y_W position in the global frame is computed with the following equations:

$$\Delta X_W = \pi r\cos(\theta)(\Delta REV_L + \Delta REV_R) \tag{2.25}$$

$$\Delta Y_Y = \pi r\sin(\theta)(\Delta REV_L + \Delta REV_R). \tag{2.26}$$

The change in X_W and Y_W are then summed in the same manner as θ to determine the X_W and

Y_W position. Finally, the velocities in the X_W and Y_W frame are computed by dividing the X_W and Y_W position values by the time separation between each time step.

2.3 Matlab Implementation

To test the validity of the equations derived with Lagrangian mechanics in 2.2.4 (Equations 2.20, 2.21, and 2.22), the trajectory and velocity profiles from a Gazebo model and a physical platform are compared to those calculated by the model in the computer program Matlab [25]. Figure 2.5 depicts the steps required to use the code and compare the dynamic model with different sources of data. [1]

The model had the same robot parameters and moments of inertia as inputs provide a useful comparison. The equations of motion derived with Lagrangian mechanics are dependent on the constant mass and inertia values as well as the variables F_x, F_y, τ and time. F_x, F_y, and τ are estimated for known angular velocities of the left and right wheel of a robot. However, this parameter is typically not directly available from a real-world platform and therefore a series of calculations is required. Our prototype platform records wheel rotational data with onboard encoders, and these data are first converted to wheel angular velocities, then to forces. Calculations associated with our prototype physical platform are found in Section 3.3.1 while the remaining calculations are found in the following subsection.

2.3.1 Calculating Force and Rate of change of Robot Heading

For a two wheeled robot, the longitudinal and lateral forces experienced by the robot chassis can be related to the wheel angular velocities, $\dot{\omega}_L$ and $\dot{\omega}_R$, the inertia of each wheel and the wheel radius, r. The dynamic equation for angular velocity of a wheel can be written as $\dot{\omega} = \frac{[T_e - T_b - r(F_t + F_w)]}{I_w}$, where T_e and T_b are the engine shaft torque and brake torque, F_t is the tractive force and F_w is the wheel static friction [26]. Our platform does not have braking capabilities so $T_b = 0$, and the shaft torque is treated to be zero. The combined tractive force and wheel static friction of the wheel is treated as the total force exerted by the wheel as shown in Figure 2.6. Therefore, the left force and right force are calculated as the following:

$$F_L = \frac{\dot{\omega}_L I_w}{\rho} \tag{2.27}$$

---

[1] All Matlab files are available at `http:\\faculty.nps.edu\thchung`

$$F_R = \frac{\dot{\omega}_R I_w}{\rho}. \tag{2.28}$$

These forces are both considered to be perpendicular to the axis of the wheels and therefore the total force in the lateral direction, F_y is zero and the total force in the longitudinal direction can be written as $F_x = F_L + F_R$. Finally, the rotational velocity of the platform, τ is simply $\tau = \frac{b}{2}(F_R - F_L)$. These calculations are done for each time step in Matlab code (Lines 258-280) then imputed into the derived equations of motion via ODE45 as described in the following section.

2.3.2 Numerical Integration using MATLAB's `ode45` Solver

To produce a visual representation of the system of equations' trajectory and speed, the equations were rearranged to form a system of ordinary differential equations, then solved. The ordinary differential equation were solved with the Matlab function ODE45, one of Matlab's differential equation solvers that is based on an explicit Runge Kutta method of integration called the Dormand-Prince pair [27]. With the specified tolerance, the solver selects a point and takes the derivative of the function at a given point. If the solution is out of the tolerance, then the step size is varied. The function requires inputs of a system of differential equations, the time span (`tspan`) , and initial conditions (`y0`). The outputs are an array of `yout` with corresponding times in a vector `tout`. In our case, ODE45 was given conditions `y0` based on initial conditions as well as values of F_x, F_y and τ associated with the specified time span. Subsequent iterations took the `yout` of the previous run of the function in order to provide a continuous trajectory.

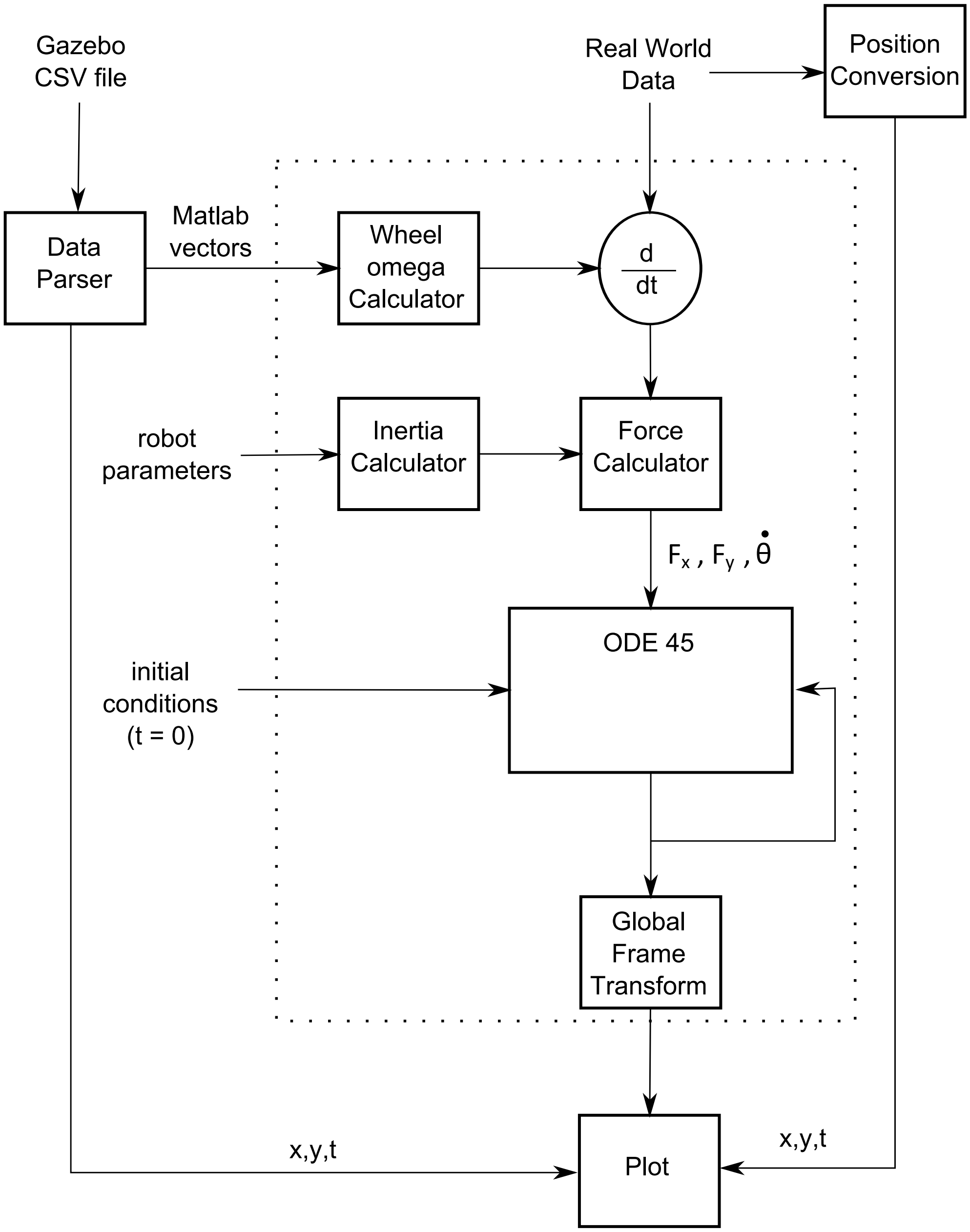

Gazebo CSV file
Real World Data
Position Conversion
Data Parser
Matlab vectors
Wheel omega Calculator
d/dt
robot parameters
Inertia Calculator
Force Calculator
F_x , F_y , $\dot{\theta}$
ODE 45
initial conditions (t = 0)
Global Frame Transform
x,y,t
Plot
x,y,t

Figure 2.5: Block diagram for Matlab model program. Wheel encoder data, robot parameters and initial conditions are input into the main file, then estimations are calculated and plotted with true data.

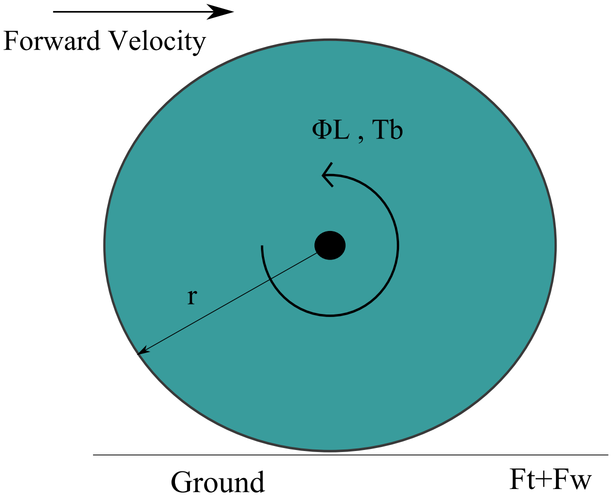

Forward Velocity
ΦL , Tb
r
Ground
Ft+Fw

Figure 2.6: This wheel force diagram represents the relationship between wheel angular velocity and the forward force exerted by the wheel, $F_t + F_w$. In the case of a two wheeled robot these forces can be combined to determine the total force in the x, y and θ directions.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Simulation and Experimental Setup

To properly provide input data and verification of performance for the model, a combination of experimental and simulated configurations are created. Using a variety of patterns from both physical and simulated platforms, raw odometry is created and processed for input into the model for later comparison against simultaneously captured motion and position tracking of the platforms.

3.1 Drive Patterns for Testing

Dynamic C code with encoder data logging is written for the physical platform to run in several geometries including a straight line, a square, a spiral and a square-spiral-line combination as shown in Figure 3.1. These patterns are created to thoroughly test the ability of the model to predict a variety of geometries that could be expected during a vehicle's navigation.

The straight line pattern assess the math models' capability to accurately predict forward speed and the total forward distance travelled while neglecting any change in heading or orientation. The robot was accelerated from zero velocity to a pre-determined speed while the wheel encoder data was recorded.

The square pattern allows for the additional analysis of the model's abilities to accurately predict the angular rotation of the platform. The square uses constant turn rates while maintaining steady forward motion. The pattern is calibrated to be 90 degrees to allow for greater ease in measuring the motion of the platform.

The spiral pattern, while similar to the square in incorporating angular rotation, but instead creates a continuously changing rate of turn to test the model's ability to handle a variety of turn rates.

Finally, a complex pattern is created to combine the various aspects from the prior patterns into a challenging test of the models ability to predict motion at the extremes of expected requirements. After a square pattern is completed, an in-place pivot turn is executed to 180 degrees followed by an opposite turn direction tightening spiral. At the conclusion of the spiral, a straight line but accelerating pattern is performed before the conclusion of the overall pattern.

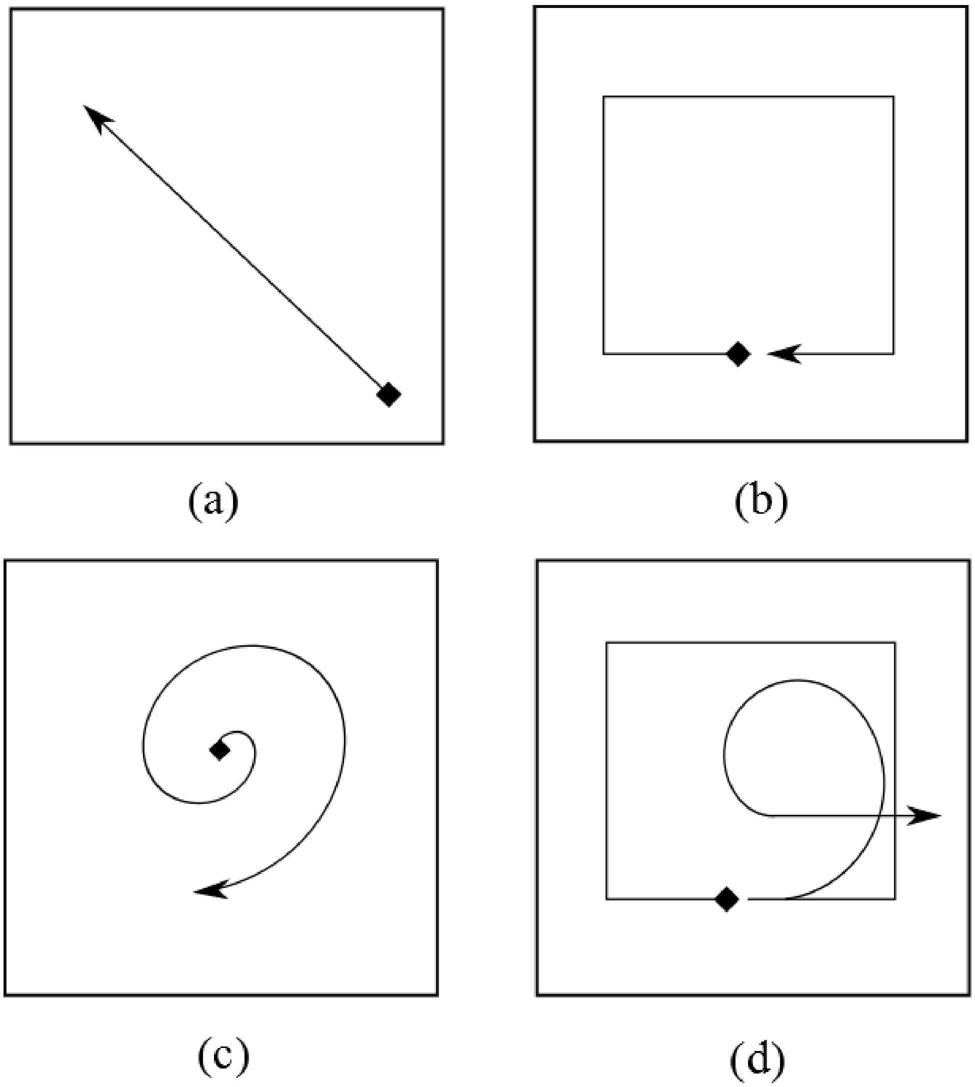


Figure 3.1: Sample patterns for use in data collection and model testing, start and end points are indicated by a diamond and arrow, respectively.

3.2 Simulation in ROS

For the necessary software and simulation requirements of modeling a vehicle in the surf zone environment, ROS was chosen for its open and flexible architecture, large user support base and future capability for advanced autonomy, vision processing and control in both simulated and physical platforms. Utilizing a graph architecture, ROS is broken down into a series of topics, nodes, messages and services, as shown in Figure 3.2, that communicate and process information between the various components necessary to monitor, control and operate a robotic system [28]. Utilizing packages and stacks, additional features and capabilities can be added to

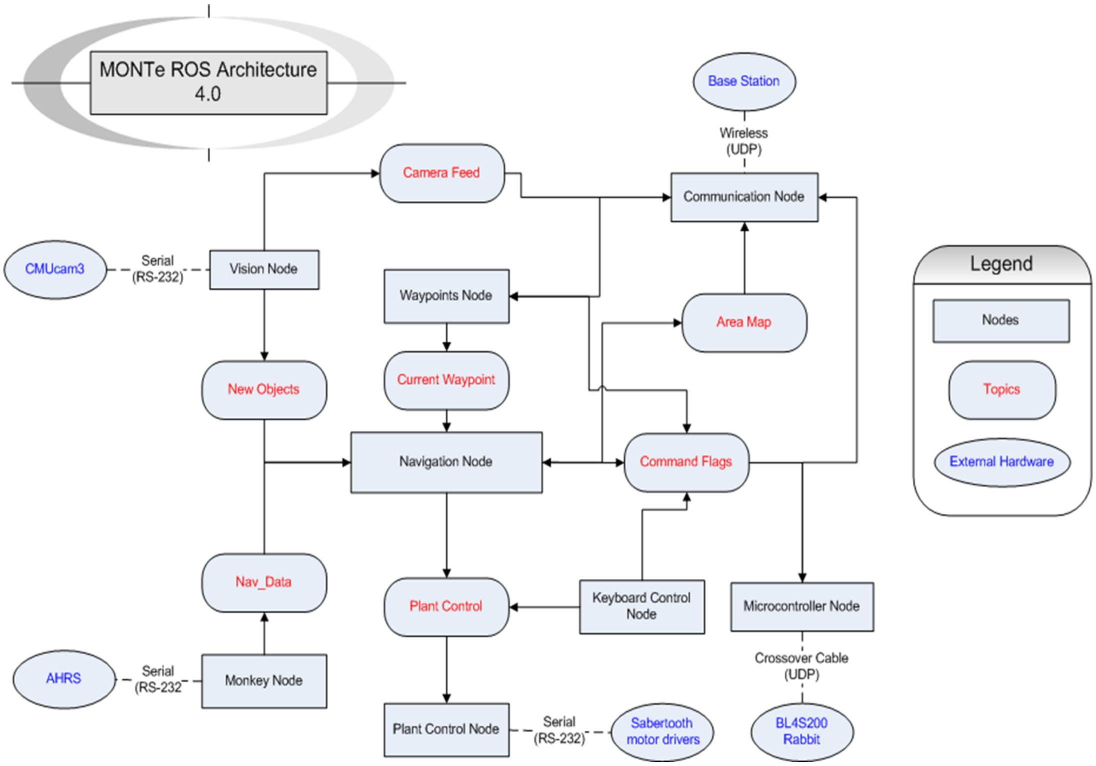

MONTe ROS Architecture
4.0
Base Station
Wireless
(UDP)
Camera Feed
Communication Node
CMUcam3
Serial
(RS-232)
Vision Node
Waypoints Node
Area Map
Legend
Nodes
Topics
External Hardware
New Objects
Current Waypoint
Navigation Node
Command Flags
Nav_Data
Plant Control
Keyboard Control
Node
Microcontroller Node
AHRS
Serial
(RS-232
Monkey Node
Crossover Cable
(UDP)
Plant Control Node
Serial
(RS-232)
Sabertooth
motor drivers
BL4S200
Rabbit

Figure 3.2: An example of the node and topic data flow for the MONTe platform. Through a series of subscribers and publishers to standard messages, various hardware and software components of the system interact in a simple and scalable architecture. From [11]

provide tools such as simulation, sensor data visualization, path planning, and mapping. This are all vital tools for later physical implementation and a critical capability that ROS provides for future work.

3.2.1 Initial Hardware Setup

A hardware configuration is assembled using standard commercial computer components or pre-made systems. The level of complexity and speed expected for the system should play as the critical role in the selection of components. Simulation is particularly demanding on the system and is the critical role for our particular test system. As such, the following components are utilized:

- Intel Core i7 Ivy Bridge Processor
- Intel Z77 Motherboard
- NVIDIA GTX 670 Graphics Card with 2GB Memory
- 128GB Solid State Drive (SSD) for Operating System (OS)

- 16GB DDR3 1600 Memory
- 750 Watt Power Supply
- Standard ATX Case

This system represents a high end configuration but much lower end systems can successfully run ROS at the sacrifice of processing and graphics speed. If the usage of ROS is limited to physical hardware control and on-board processing, a much more minimal system configuration will suffice. Integrated small form-factor computers such as the Stealth PC [29] are ideal for their processing power and input/output capability and shown to be successful on prior platforms [12].

3.2.2 Initial Operating System Setup

The only fully supported OS for ROS is Ubuntu. The Linux interface, command line terminal windows, and root access are all critical features found in Ubuntu [30]. A large release history is available and each version is compatible with only specific versions of ROS [14] and compatibility testing is necessary. The test system built utilizes the following compatible software configurations:

- Ubuntu Narwhal 11.04 with ROS Diamondback
- Ubuntu Ocelot 11.10 with ROS Electric
- Ubuntu Pangolin 12.04 with ROS Fuerte

ROS releases follow an alphabetical naming scheme similar to Ubuntu. Diamondback, Electric, and Fuerte are the current stable releases with Groovy being the latest planned release. Of particular concern with the compatibility testing is successful video driver support. Third-party and experimental drivers are needed for the test system and are the main reason other software configurations can not be implemented.

3.2.3 Package and Platform Installation

The active user base for ROS provides an ever expanding selection of code, packages and stacks for implementation. Depending on the expected use, the most advantageous packages are added. The test system uses the following packages for the listed capabilities:

- **Gazebo:** Installed by default with the ROS Full Desktop Installation, provides visual simulation of robot platforms and 3D worlds.

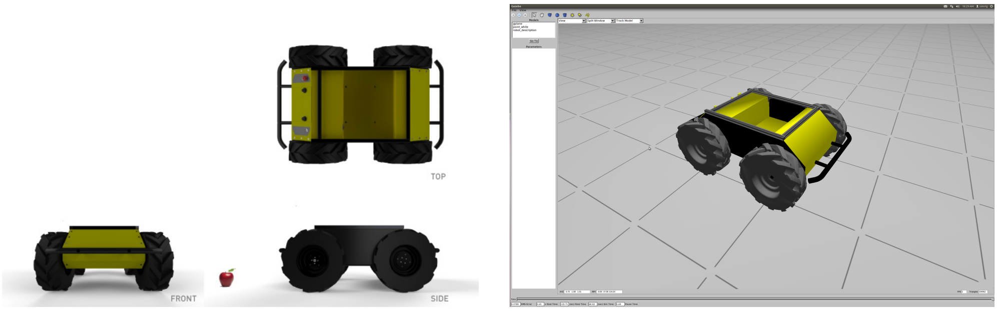
- **RVIZ:** Installed by default with the ROS Full Desktop Installation, provides visualization of raw physical and simulated sensor data, object detection, path planning and mapping.
- **iRobot Create:** A collection of drivers, tools and robot model for simulation of the iRobot Create platform [31].
- **laser-drivers:** A collection of drivers for the use of scanning laser range finders and LIDARs from manufactures Hokuyo and SICK.
- **NPS ROS:** A collection of code, drivers and prepared files for launching and interacting with a variety of vehicles in both physical and simulated use.
- **Clearpath Husky:** A highly supported robotic vehicle with high resolution models, drivers and tools for launching and interacting in both physical and simulated use [16].
- **Hector Gazebo Worlds:** A Gazebo package that includes high quality mesh and world files for the setup and creation of realistic terrain.

3.2.4 Gazebo Model

Simulation is performed by the Gazebo package for ROS. Utilizing the Open Dynamics Engine and OpenGL rendering, it provides high quality 3D visualization of robotic systems in complex worlds [14]. Gazebo serves the roll of physical motors, actuators and sensors interacting with terrain and obstacles, providing the rest of the ROS system the same subscription and publishing of messages that would be taking place if a physical robot were connected and placed in a real environment. A large amount of flexibility is available for the configuration of simulated platforms, sensors, terrain features, obstacles and dynamic motion.

3.2.5 Robotic Vehicle Models

A large number of developed robotic vehicle models are available for use. They are built from uniform robot data files (.URDF) and constructed from simplified shapes, panels and components arranged from a base coordinate frame through parent and respective link transformations. Each component of a model is assigned a full inertia tensor and given free, rigid or restricted motion relative to its parent link. When fully combined, the model interacts with realistic physics and motion respective of a real physical platform of the same construction. Drivers are written to translate standard twist and odometry messages into motion respective to the particular drive configuration of the robot. For our purposes, two vehicles are chosen to represent the approximate size shape and drive configuration of past successful and experimental NPS platforms as well as the considerations of the CONOPS. The iRobot Create and Clearpath Robotics Husky A200, as shown in Figure 3.3, are selected thanks to their detailed construction, similar



(a) (b)

Figure 3.3: (a) Profile drawings of the Husky A200 platform. From [16], (b) A 3D perspective view of the Husky .URDF file spawned in the Gazebo simulation software.

configuration to the eventual physical test platform, and availability as a proven and supported commercial robotic system for potential physical testing in the future [16], [31].

3.2.6 Creating simulated data

With all hardware and software components assembled, simulated worlds and models are spawned utilizing .xml launch files with the following commands for example:

```
roscore
roslaunch gazebo_worlds empty_world.launch
roslaunch husky_description base.urdf.gazebo.launch
```

Each command executes a series of operations. The command **roscore** starts the main server that handles all messages and subsequent ROS packages. The command **roslaunch gazebo worlds empty world.launch** spawns the default flat empty world in the Gazebo simulation software. Finally, the command **roslaunch husky description base.urdf.gazebo.launch** spawns the necessary components and controllers into the Gazebo simulation software. Due to the assumptions of the mathematical model to a two-dimensional (2D) plane, a flat and featureless world is spawned and used as a test area. Robotic models are controlled through either manual tele-operation or pre-determined scripts. Data is exported in real time to include:

- Simulation Time
- Left Wheel Angular Velocity
- Right Wheel Angular Velocity
- Robot Heading

- Robot Position in *X*
- Robot Position in *Y*

This data is saved in a space separated value (.ssv) format and used as the input and ground truth comparison in the mathematical model.

3.3 Physical Test Platform

To provide sources of data from real-world sources, a physical platform, as shown in Figures 3.4 and 3.5, is constructed for the intentions of test runs on several types of surfaces and terrains. From experiences with past NPS surf zone robots, special emphasis is placed on a simple and durable platform. Constructed in-house by the physics department research staff, the test platform utilizes milled aluminium and pneumatic tires with no dedicated shock absorption system. Servo motors are used through a planetary gear box and chain drive reduction allowing for a compact and high torque drive configuration. Each servo motor has a ten pole quadrature magnetic encoder mounted inline with the motor shaft. Motor controls are handled by an onboard BL2000 Rabbit Processor. Motor commands are sent via hexidecimal over RS-232 to an adapter board where it is converted to RS-485 and passed to the four independent servo controllers.

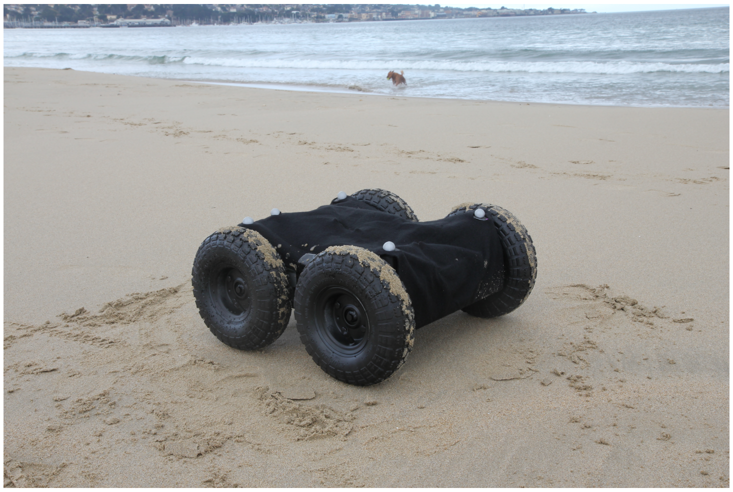


Figure 3.4: Photograph of four wheel prototype platform used for physical testing.

The test platform allows for rudimentary skid-steered analysis and comparison to the developed models. The four-wheeled platform was built with a simple and reliable design to replicate data collection. Tire wheels are chosen over the normally used Whegs$^{\text{TM}}$ in prior NPS surf zone

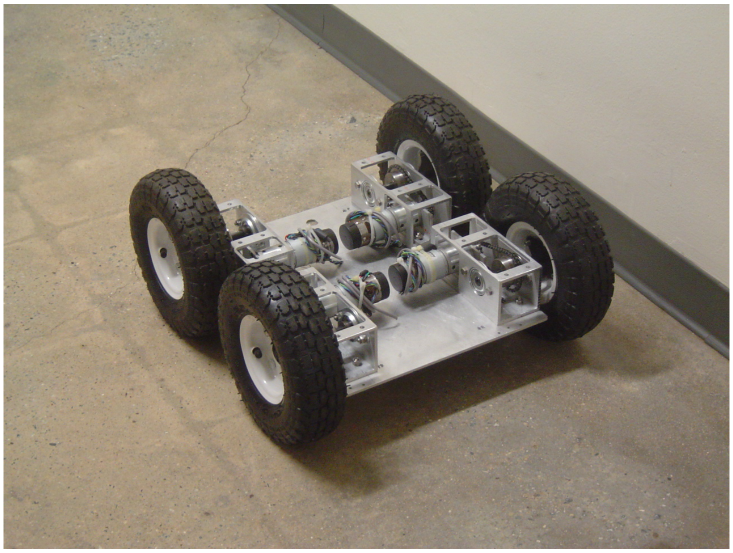
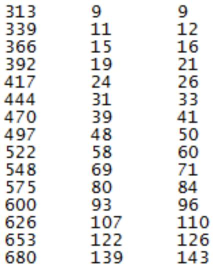


Figure 3.5: Photograph of test platform interior construction with servo motor and gear reduction layout.

projects. This is done in order to better match the model, although both the model and physical platform can later be adapted for Wheg$^{\text{TM}}$ inclusion.

3.3.1 Creating Physical Platform Data

The Matlab script requires inputs of angular velocity of each wheel in addition to time, initial conditions, and other constant physical parameters in order to predict the robot's trajectory. The angular wheel velocities are not directly available from the encoders, but can be interpolated with a series of calculations. The data from the physical platform encoders are written to a text file similar to that depicted in Figure 3.6 containing the time in milliseconds, and the port and starboard wheel positions. Next, these data are separated into three Matlab variables identified as `ttest`, `portpos` and `stbdpos` for the time, port position, and starboard position, respectively. Since the encoders are taking readings directly off of the drive shaft of the motor, the encoder ticks per revolution, planetary gear box ratio and chain drive reduction are divided out of the values. The `portpos` and `stbdpos` vectors are thus normalized to the fraction of wheel revolutions completed by the left and right wheel at a corresponding time. Finally, the angular wheel velocity of the left and right wheels in radians per second is approximated to be $\dot{\theta} \approx \frac{\Delta\theta}{\Delta t}$ where θ is the change in angle of the wheel between time t and time $t+1$. The Matlab steps described in this section are shown in Figure 3.7.



```
313     9      9
339     11     12
366     15     16
392     19     21
417     24     26
444     31     33
470     39     41
497     48     50
522     58     60
548     69     71
575     80     84
600     93     96
626     107    110
653     122    126
680     139    143
```

Figure 3.6: A Sample of physical platform encoder data text file. The columns from left to right are: time in milliseconds, left wheel encoder count, and right wheel encoder count.

```
data = importdata('14foot6inchSquare.txt'); %Square run text file

ttest = data(:,1);
portpos = data(:,2);
stbdpos = data(:,3);

tspanentire = 1E-3.*ttest'; %Converts from ms to s
wheelrevport = (portpos')./(98*20); %Computes total number of wheel
    revolutions
wheelrevstbd = (stbdpos')./(98*20);
wL = (2*pi).*[0 diff(wheelrevport)]./ [1 diff(tspanentire)];
                 % Computes wheel angular velocities in rad/sec
wR = (2*pi).*[0 diff(wheelrevstbd)]./ [1 diff(tspanentire)];
                 % First element set to zero to indicate start w
```

Figure 3.7: A Sample of Matlab code for the conversion of physical platform encoder data to wheel position, then to wheel angular velocity. Note that ‘14foot6inchSquare.txt’ is a example file name for the physical platform data file.

3.3.2 Motion Capturing Setup

From the test patterns described, an accurate method of tracking the vehicles motion in 3D space is required. For simple geometries such as the straight line and square geometries of constant velocity, simple tape measure and stop watch data collection is sufficient to verify that the model produces results with the correct X and Y distance and approximated speed. However, for the more complex patterns such as the spiral and combination pattern, a different approach is required. Specific interests in parameters such as position, velocity, angular acceleration, and heading as a function of time are critical comparisons for higher level analysis of the model.

For complex motion tracking, the Vicon optical tracking system is chosen [32]. The system is housed at NPS and consists of a large bay, approximately 10 meters by 10 meters, and surrounded by eight Vicon T160 cameras and integrated infrared illuminators, as shown in Figure

3.8. Each camera is capable of capturing 16 megapixel resolution (4704 x 3456 pixels) at frame rates of 120 frames per second. This data is gathered and processed by the Vicon system and displayed in 3D space, providing three coordinate translational position and four coordinate quaternion rotation at 100 hertz with sub-millimeter accuracy. The only modification to the test platform to utilize this motion tracking is the addition of reflective markers to the topside of the platform. They are placed in a non-symmetrical pattern and used by the Vicon system.

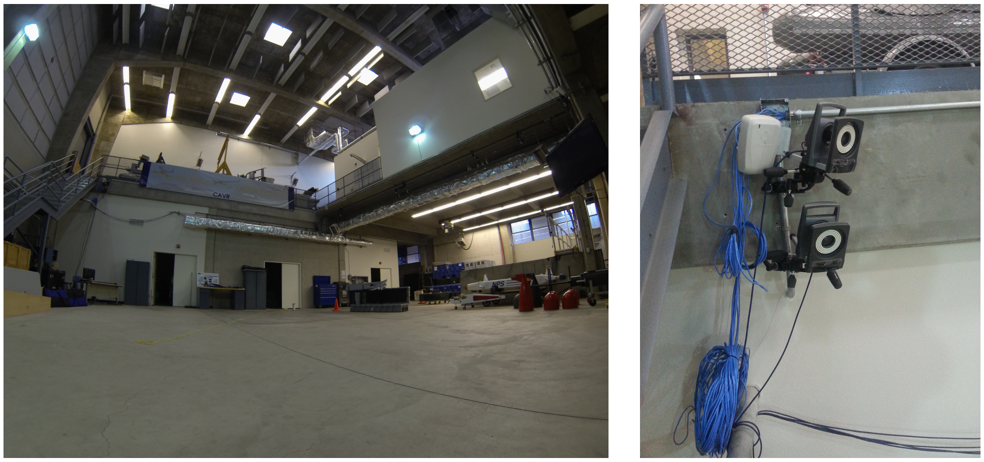

(a) (b)

Figure 3.8: A picture of the Vicon motion tracking lab in Halligan basement housing eight cameras. The combination of high precision accuracy and large open space is an ideal combination for large scale motion analysis. (b) Vicon T160 motion capture cameras with built in infared illumination are capable of 16 megapixal resolution at several hundred frames per second.

The Vicon system outputs a ROS .bag file containing positional (X,Y) and angular (θ) data that was then adapted into a .txt file and loaded by the Matlab script.

The complex square-spiral geometry was saved as a file named `20121011113608bag.txt`. The time vector was normalized to run from $t = 0$ to $t = t_f$, then the X and Y position was plotted as a function of time. The X and Y velocity in the global frame as well as the platform's forward velocity in the robot frame were calculated as shown in Figure 3.9.

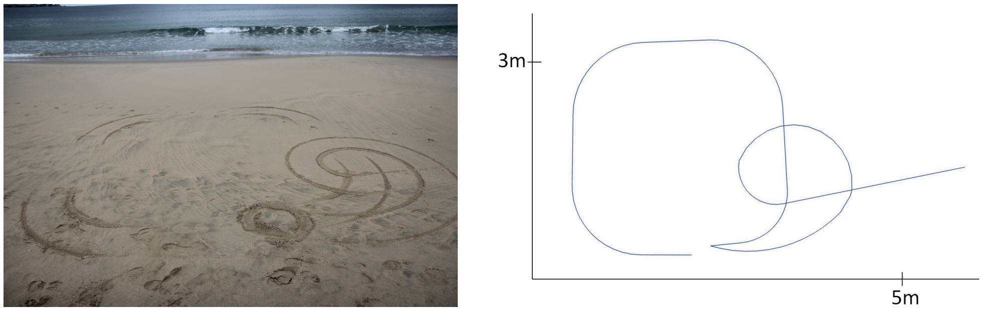
3.3.3 Beach Testing

With surf zone operations the ultimate goal of the platform, specific testing is performed on various beach sand environments. The local Del Monte beach in Monterey, California serves well for this purpose. In order to approximate the robot's true position in a sandy beach environment, a simple Cartesian coordinate system is laid out and data points are taken of all important

```
for j = 2:length(tbagnormalized)
    bagXdiff(j) = (bagX(j) - bagX(j-1)); %Change in X Position (
       Global)
    bagYdiff(j) = (bagY(j) - bagY(j-1)); %Change in Y Position (
       Global)
    xdistance(j) = sqrt( bagXdiff(j)^2 + bagYdiff(j)^2); %Distance
       traveled in one time step
    tbagstep(j) = tbagnormalized(j) - tbagnormalized(j-1); %Time
       between readings
    bagXdot(j) = bagXdiff(j)/tbagstep(j); %X Velocity (Global)
    bagYdot(j) = bagYdiff(j)/tbagstep(j); %Y Velocity (Global)
    bagxdot(j) = xdistance(j)/tbagstep(j); %Foward Velocity in robot
        frame
    bagxdotvector = [bagxdotvector bagxdot(j)]; %Forward Velocity
       vector
end
```

Figure 3.9: Matlab Code for the conversion of '.bag' file positional data (*bagX*, *bagY*) to velocity (*bagXdot*, *bagYdot*) and forward velocity in the robot frame (*bagxdot*)

features found on the marked track laid out by the vehicles wheels as shown in Figure 3.10(a). This data is then input into a graphics program where overall motion lines are interpolated and drawn as shown in Figure 3.10(b). Overall results of this data collection means are accurate and reliable for the purposes of comparison between real motion and model prediction.



(a) (b)

Figure 3.10: (a) Photograph of hard pack sand trajectory taken at Del Monte Beach. The track marks are used to record the driven motion in a simple Cartesian coordinate system. (b) The pattern reproduced from driving on hard sand. Slip was minimal except for very tight turns and the motion is very similar to calibrated patterns on concrete.

This technique does not allow us to approximate the position as a function of time nor does it give us the linear and angular velocity data present in Vicon based tracking. However, our current prototype physical platform does not have GPS capabilities and the motion is on a small enough scale that GPS data would be of insufficient accuracy. Therefore this technique was deemed satisfactory and sufficient for position estimation.

CHAPTER 4:
Results

The simulated and physical platform are run in several different geometries, including a straight line, a spiral, a square pattern, and a square-spiral-line combination. The directly exported wheel angular velocity or encoder text files are recorded and input into the dynamic and kinematic Matlab models. In addition, precise position data from ".bag" files recorded from the Vicon system or position data from simulation are recorded during the same experiments are inputted into Matlab for comparison with the model results. The Vicon data simulated position data are considered to be ground truth for the purposes of these experiments. Details of results and analysis for the straight line, square, and square-spiral-line patterns are presented the following sections. The spiral pattern results are omitted from chapter because the metric of performance tested in this pattern is already assessed during the square-spiral-line combination run.

4.1 Validation of Model with ROS Simulation

For initial verification of the model performance and accuracy, a simple two-wheeled differentially steered platform was simulated in ROS. The iRobot Create, discussed in Section 3.2.5, matches these requirements and will serve this purpose. The vehicle is given a straight line path in the form of a single waypoint five meters directly in front of its initial heading. Then an approximately square pattern is given to drive using a series of waypoints and a potential field driver. Data is recorded and imported into Matlab where the model overlays its predicted motion given the same angular velocities of the wheels. For all of the graphical representations in this section, the color red denotes the recorded position in simulation while blue is the dynamic results. All results are presented in SI units, with the exception of angular measurements in degrees.

4.1.1 Results and Analysis of Simulation and Dynamic Model Comparison

The dynamic model prediction shows excellent correlation to the recorded track of the robot from the ROS simulation. As shown in Figures 4.1 and 4.3, a light underestimation of the linear velocity is made for both the straight line and square pattern. This results in a shorter linear distance traveled, as shown in Figures 4.2 and 4.2. This success provides strong confidence in the model's accuracy and further application to physical platform testing.

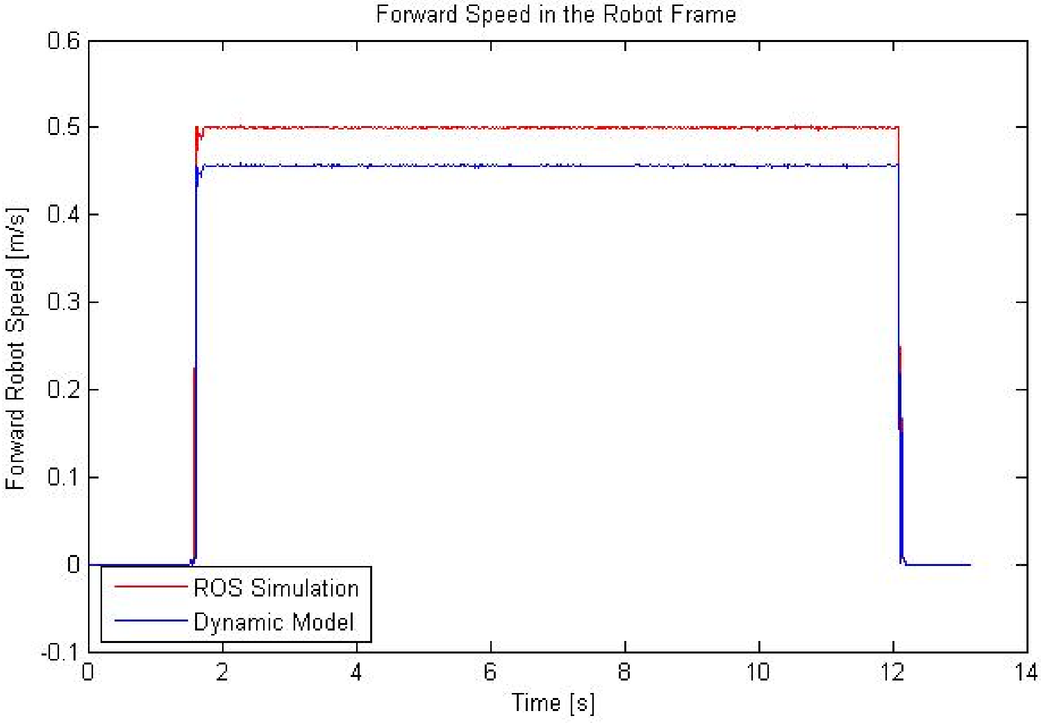


Figure 4.1: Forward velocity comparison of ROS simulation and dynamic model results for a straight line run. Simulation velocity is shown in red and the models predicted velocity is shown in blue. The Model slightly under-predicts the velocity of the simulation platform.

4.2 Physical Platform Results

After seeing strong correlation and excellent success in the models ability to predict the motion of a simulated platform, we move on to real world data and model prediction with the physical test platform discussed in Section 3.3.

4.2.1 Straight Line Pattern

A simple straight line experiment in which the velocity is put through a trapezoidal pattern is conducted with the physical platform. From rest, the vehicle is accelerated to a fixed velocity then decelerated to rest in order to reach a distance of approximately three wheel revolutions. The left and right wheel encoders verified that the position of the wheels were symmetric across time. The Vicon motion capture, kinematic, and dynamic model results are presented in Figures 4.5, 4.6, and 4.7. Figure 4.5 shows the robot's position in the global coordinate frame as a

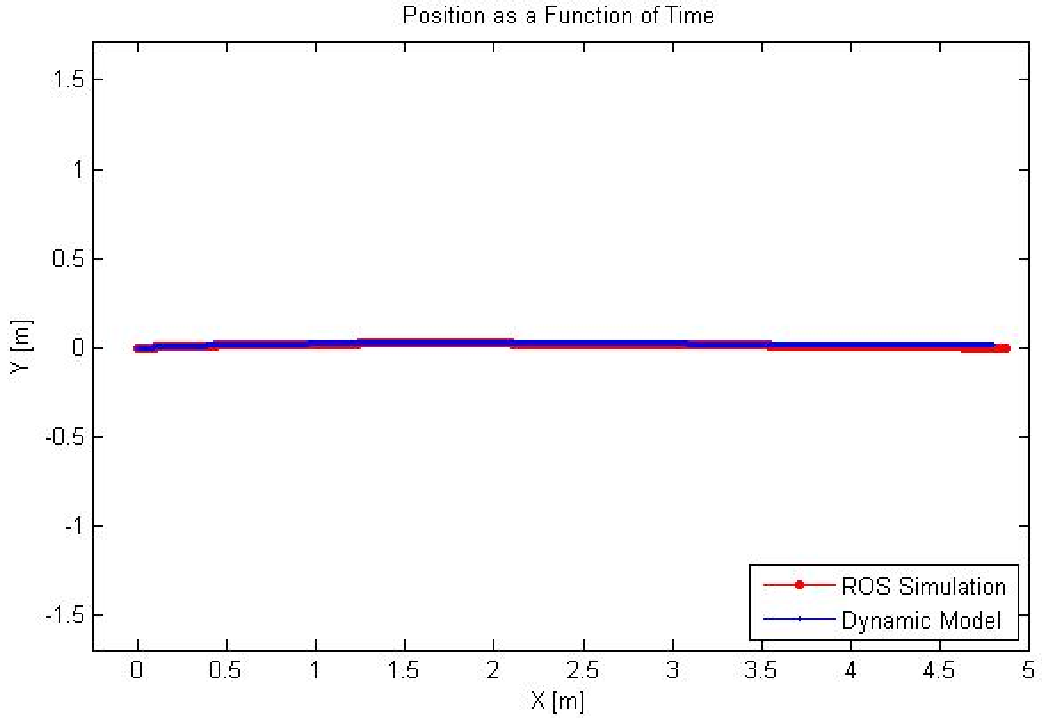


Figure 4.2: Trajectory comparison of ROS simulation and dynamic model results for a straight line run. As expected from the slight under-prediction of velocity, the final distance traveled by the simulated platform is slightly farther than the models prediction.

function of time. For all of the graphical representations, the color red denotes the motion capture results while green and blue are the kinematic and dynamic results, respectively. All results are presented in SI units, with the exception of angular measurements in degrees. The robot begins at the origin and moves forward at a heading of approximately 136 degrees. For this run, the motion capture results show the total distance traveled in 5.52 seconds to be 2.51 meters. The final estimated position of the kinematic and dynamic models is off by 6 and 4.7 centimeters, respectively, when compared to the motion capture trajectory. The motion capture results are treated as ground truth and therefore the dynamic and kinematic models slightly under-predict the total distance traveled. This under-prediction is caused by both models under-prediction the forward speed of the robot.

In Figure 4.6, the forward speed of the robot is shown as a function of time for the five second of this run. The speed increases from zero to a steady state of approximately 0.62 m/s, then ramps down to zero near the end of the run. Variation in the steady state speed is observed for all three results. Despite the sub-millimeter positional precision of the Vicon system, there is

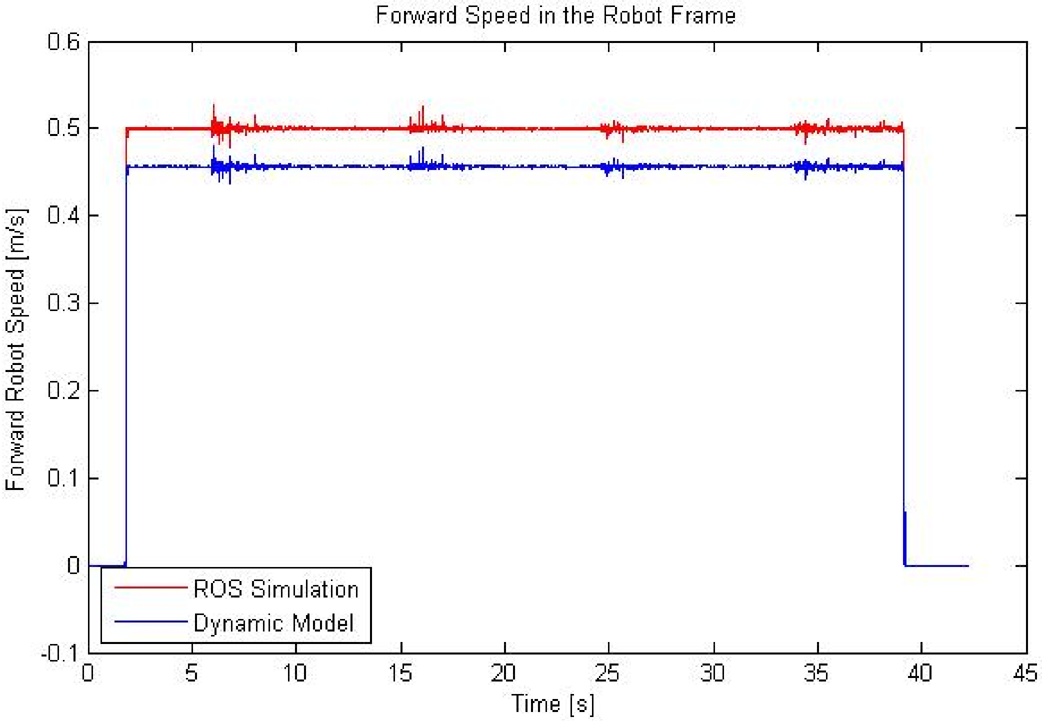


Figure 4.3: Forward velocity comparison of ROS simulation and dynamic model results for a square pattern. Simulation velocity is shown in red and the models predicted velocity is shown in blue. The Model slightly under-predicts the velocity of the simulation platform.

variation in the forward speed. This variation is likely due the numerical differentiation, $\frac{\Delta x}{\Delta t}$, in determining the robot's velocity from location data. The variation in the kinematic and dynamic results appear to follow the same trends and are due to noise associated with the wheel encoders and the limitations of the physical platforms motor control system. The forward speed differs most notably between the models and the motion capture results during the acceleration and deceleration periods.

Figure 4.7 is a compilation of graphical results for the same straight line run. The first and second graph show the X position (X) and X velocity (v_x) as a function of time. The third and fourth graphs show the position and velocity as a function of time for the Y axis. The kinematic and dynamic model results are nearly identical for the straight line run, with variations due to encoder noise and physical platform limitations. The X and Y velocities are again observed to differ most notably during the acceleration and deceleration periods. Finally, the fifth and sixth graphs in this figure depict the heading and angular velocity of the robot as a function of time. Again, the kinematic and dynamic model results are nearly identical. The code was written

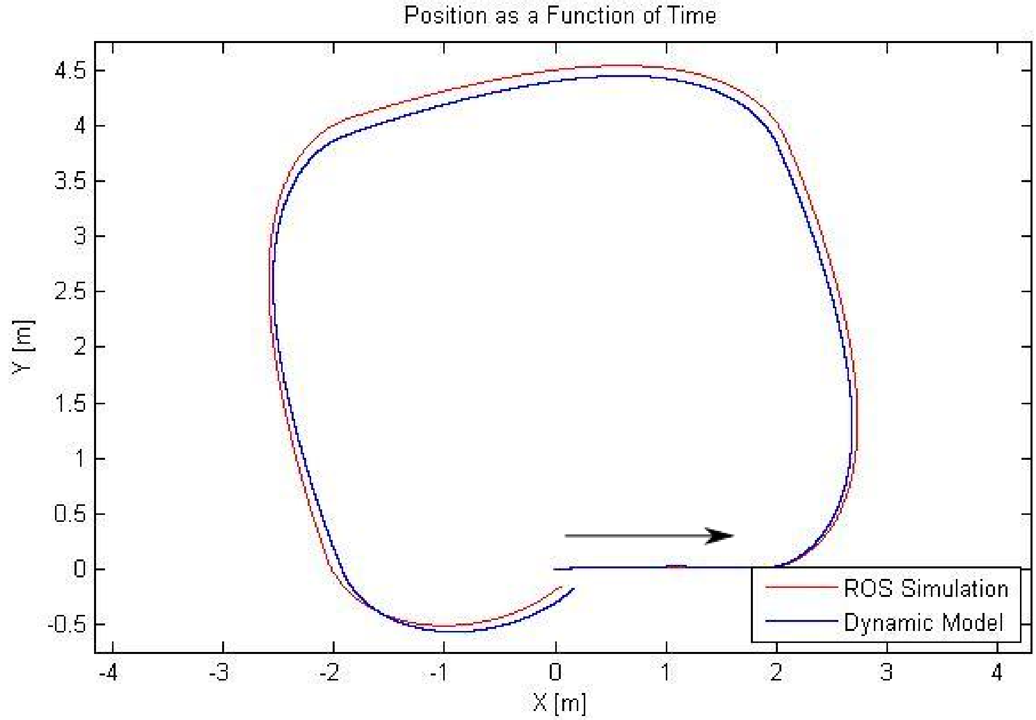


Figure 4.4: Trajectory comparison of ROS simulation and dynamic model results for a square pattern. The arrow indicates the direction of motion. Simulation velocity is shown in red and the models predicted velocity is shown in blue. As expected from the slight under-prediction of velocity, the predicted motion by the model creates a slightly smaller square than the path traveled by the simulated platform.

to command identical motor signals for both the left and right wheels, however, in reality the motors are given slightly different commands due to noise. These differences were observed to be approximately between one and two encoder ticks per revolution. Additionally, the encoders experience slight noise inaccuracy. The compound effect of these variations explains why the models predict that the trajectory be nearly straight, but will have slight error. The Vicon data indicate that the robot does not completely maintain its initial heading but instead experiences a decrease in heading. This is supported by our observation that the robot tracked slightly to the right due to differences in the left and right wheel. The models treat the wheels as being identical, which is why they do not depict the same phenomenon.

4.2.2 Square Pattern

The straight line pattern assess the mathematical model's capability to accurately predict forward speed and the total forward distance traveled while neglecting minimal change in heading

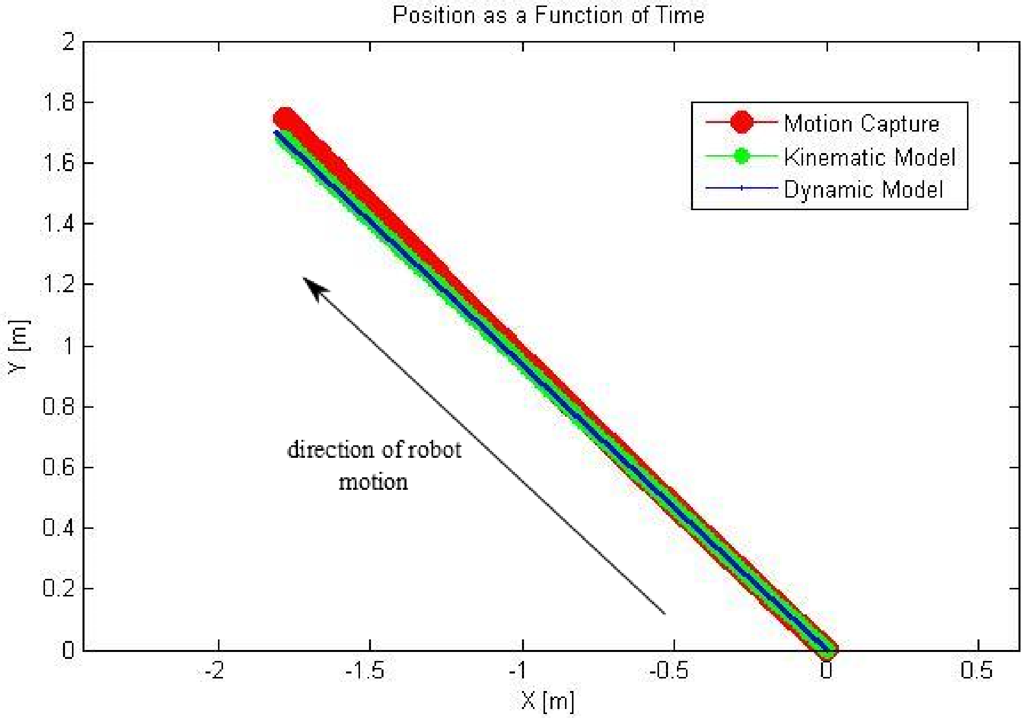


Figure 4.5: Graphical representation of the robot's trajectory in the global coordinate frame, X and Y in meters(m), as a function of time in seconds(s). The color red indicates the motion capture results while green and blue are the kinematic and dynamic models, respectively. The robot begins at the origin and moves forward with a heading of approximately 136 degrees.

or orientation. Therefore, the square pattern was selected to assess the mathematical model's ability to predict the angular rotation of the robot. The square run begins at zero velocity at the midpoint of one side of the square, with a heading of 180 degrees. The robot is then accelerated to a steady forward velocity. Near the corner of a perfect square, the Dynamic C code invokes a 90 degree turn while maintaining constant forward velocity. A "car" turn rather than a "pivot" turn is expected through this process. Upon the completion of the turn, the robot is directed to return to the same steady forward velocity, and continue in this manner until it is near the starting position.

The results for the Vicon setup are presented in Figures 4.8, 4.9, and 4.10. The motion capture data recorded the trajectory to approximate a square with rounded corners. The forward speed appears to ramp up to approximately 0.6 m/s and stays at this speed for the duration of the experiment regardless of whether or not the robot is turning or maintaining heading as observed in Figure 4.10. The total duration of this run is 26 seconds. The total distance covered is

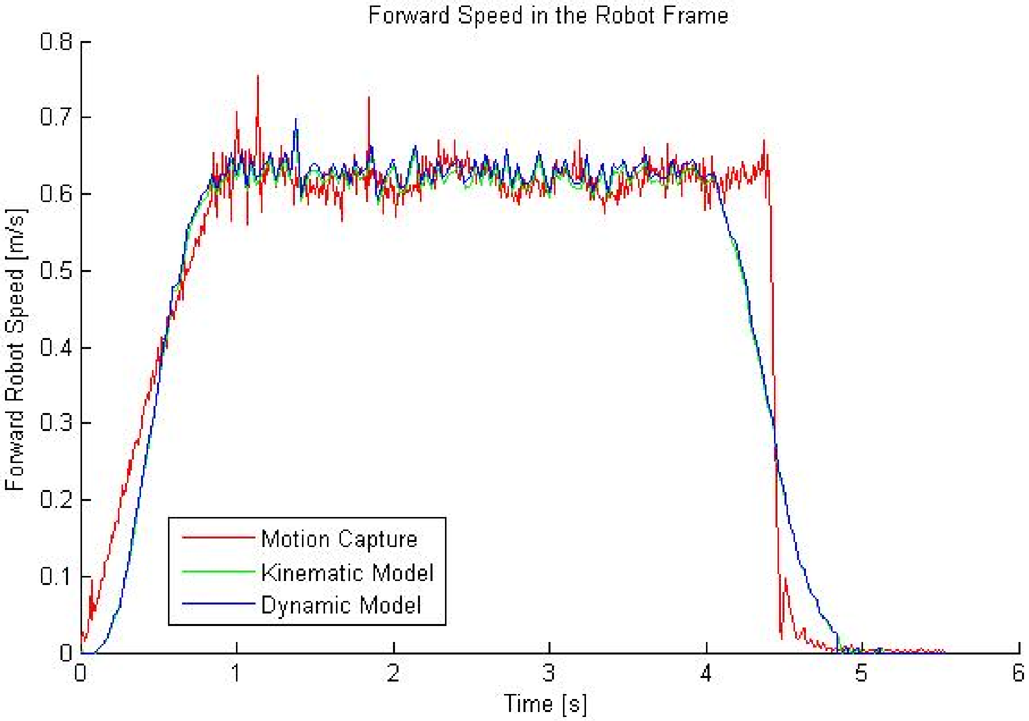


Figure 4.6: The forward speed of the physical platform, in the global coordinate frame, as a function of time for the straight line experiment.

approximated to be 15.57 meters as estimated by taking the integral of the motion capture forward speed with respect to time. The kinematic and dynamic model final positions are 48 and 4.1 centimeters in error, respectively, when compared to the motion capture final postion.

Next the results for the kinematic model are compared to the motion capture results in Figures 4.11, 4.12, and 4.13. These figures show significant discrepancies between the motion capture and kinematic model results. Figure 4.11 suggests that the kinematic model is estimating a turn of much more than 90 degrees at each corner of the square and this observation is confirmed in Figure 4.12. The average heading in leg one of the square as estimated by the motion capture data is 180 degrees, which is identical to that estimated by the kinematic model. However, the peak angular velocity during the first turn of the square is determined by the motion capture to be -30 deg/s compared to -50 deg/s by the kinematic model. The increase in angular velocity results in a smaller heading during the second leg of the square, measured to be 180 degrees by the motion capture, and only 52 degrees by the kinematic model. This error continues to propagate as shown in the figure.

The reason for this error comes from the assumptions stipulated for the kinematic model. It is derived from a two wheeled, differentially steered platform that experiences no slip in linear or rotational motion. Regardless of the wheel motion, no slip is experienced and exact amounts of turn can be directly calculated from the physical relation of the axle width and wheel diameter. The center of rotation is located directly in line with the single axle and the wheels are free to rotate with no lateral slip. When these same assumptions are put on a four-wheeled differentially steered vehicle, the derivation breaks down. A four-wheeled vehicle using a differentially steered locomotion must undergo an amount of lateral slip during turns. The location of the axles being offset from the center or rotation physically means that any rotational motion must also include lateral slipping at the contact between the wheels and the ground. A four-wheeled differentially steered vehicle is commonly called a skid-steer configuration for this very reason. As the vehicle enters a turn, the kinematic model is assuming that perfect efficiency is being translated from the differential in wheel velocities into rotational motion, while in fact a proportion of this differential is being lost to lateral slipping. Due to this physical difference, the kinematic model over-estimates the amount of turn given angular velocities of the wheels.

The heading error was mitigated by scaling the the heading, θ, at each time step for the kinematic model by a factor of 0.73. This corrected the trajectory as shown in Figure 4.14. Although the resulting turn angle remains slightly more than 90 degrees, this scaling correction results in a noticeable improvement in accuracy. The improvement due to this scaling can also be observed in the alignment of the heading and angle results as shown in Figure 4.15.

The results of the dynamic model are compared to the motion capture and kinematic model results in Figures 4.16, 4.17, and 4.18. The dynamic model's total inertia (I_R) and wheel inertia(I_w) are adjusted by a factor of 1.7 and 0.97, respectively. This scaling is appropriate in that an ideal model is used to initially calculate inertia for the body and wheels. The wheels were assumed to be uniform cylinders of constant density. Under closer analysis, they are pneumatic rubber tires fitted on a steel hub. A larger portion of mass is located nearer to the center of rotation and as such a better approximation of the true inertia is a slightly lower amount than the idealized form. An idealized model is also used for the body being a rectangular box of uniform density. In physical construction, the body is a machined aluminum exterior with a hollow center fitted with the necessary motors, gear reductions, servo controllers and batteries. The majority of these inner components are located near the outside perimeter with only wiring and light weight circuits comprising the center portion. Opposite of the wheels, the aluminum shell and perimeter components account for a much larger proportion of the mass as well as being farther from the center of rotation. Therefore, the true inertia is significantly higher than the idealized approximation.

Figure 4.16 shows that the dynamic model can be scaled so that the trajectory is nearly aligned with the motion capture results. Figure 4.17 shows, just as it showed for the straight line run, that the forward speeds appear to be aligned as well. Finally, Figure 4.18 shows that the dynamic model aligns well with the motion capture data.

4.2.3 Square-Spiral-Line Pattern

A series of geometric patterns were combined into one run to test a variety of different aspects of the mathematical models. After the square pattern is completed, an in-place pivot turn of approximately 180 is executed by commanding the left and right wheels with equal and opposite motor commands. Following the pivot, the robot then is commanded to begin a counterclockwise spiral of increasing velocity. Upon completion of this spiral, the vehicle is continuously accelerated forward until given the command to stop.

The results of this experiment are presented in Figures 4.19, 4.20, 4.21, and 4.22. Figure 4.19 depicts the motion capture data for this complex geometry. The run begins at the origin with a robot heading of 180 degrees, and completes a counterclockwise circle with motor commands identical to those given in the square configuration. In an ideal case, a pivot turn would result in rotational motion with zero translational motion. In reality, as observed by the Vicon system and shown in Figure 4.20, the robot experiences a small forward velocity during the pivot. As observed in 4.21, neither the dynamic nor the kinematic model accounts for this translation but rather assumes a perfect pivot turn. This results in both the kinematic and dynamic models slightly overestimating the change in heading during this maneuver, and therefore both begin the spiral portion at an incorrect heading (Need to add actual heading, possibly zoom in of this point or the end of spiral point in figure similar to squarespiralfig2all). The kinematic model final position after this 60 second, 22.16 meter run is 1.27 meters different from the motion capture position while the dynamic model is 1.02 meters different.

4.3 Advanced Testing in Surf Zone Terrain

For final verification and analysis of the model, several runs are performed on hard and soft sand. The local Del Monte Beach in Monterey, California provides this testing ground. Due to no motion tracking available in the outdoor test area, the marked sand path data collection discussed in Section 3.3.3 is used for ground truth. Also, only the complex square-spiral-line pattern calibrated for concrete is tested as it combines all of the specific criteria of the other three patterns.

4.3.1 Hard Sand

A relatively flat and uniform section of sand is chosen for this test. Specifically, a portion of sand that is submerged during high tide and exposed at low tide producing a damp and relatively high mass material. The pattern is run and wheel encoder data and drive path are recorded. The data is processed through the model with necessary force coefficients applied and the predicted path is compared to ground truth as shown in Figure 4.23.

The material exhibited similar properties to concrete during the initial square pattern with only a slight difference in starting and pivot location of 0.38 meters. In contrast, during the pivot turn, wheel slip is much more dramatic. The hard sand tends to provide good friction and minimal lateral slip during broad turns but as the radius of curvature tightens, slipping becomes pronounced. The dynamic model is unable to account for this non-linearity in slipping.

4.3.2 Soft Sand

A relatively flat section of sand is chosen for this test. A slightly more pronounced slope was present than with the hard sand run with the back length of the square slightly elevated compared to the starting location. In contrast to the hard sand, this area sees no regular exposure to water and presents a loose, dry and light weight medium. In addition, the uniformity observed in the hard sand is not present on soft sand with constant lumps, dips, and irregularities in the surface. The pattern is run and wheel encoder data and drive path are recorded. The data is processed through the model with necessary force coefficients applied and the predicted path is compared to ground truth as shown in Figure 4.24.

The material exhibited very different properties with high levels of non-linearity. The model predicts each turn as uniform in angle and length but ground truth reveals different levels of roundness and angle carried through identical turns. The amount of slip is much higher and the final position of the square is 1.8 meters from the starting position. It was observed visually during the run that the vehicle was experiencing larger amounts of vertical and non-planar motion compared to all other runs.

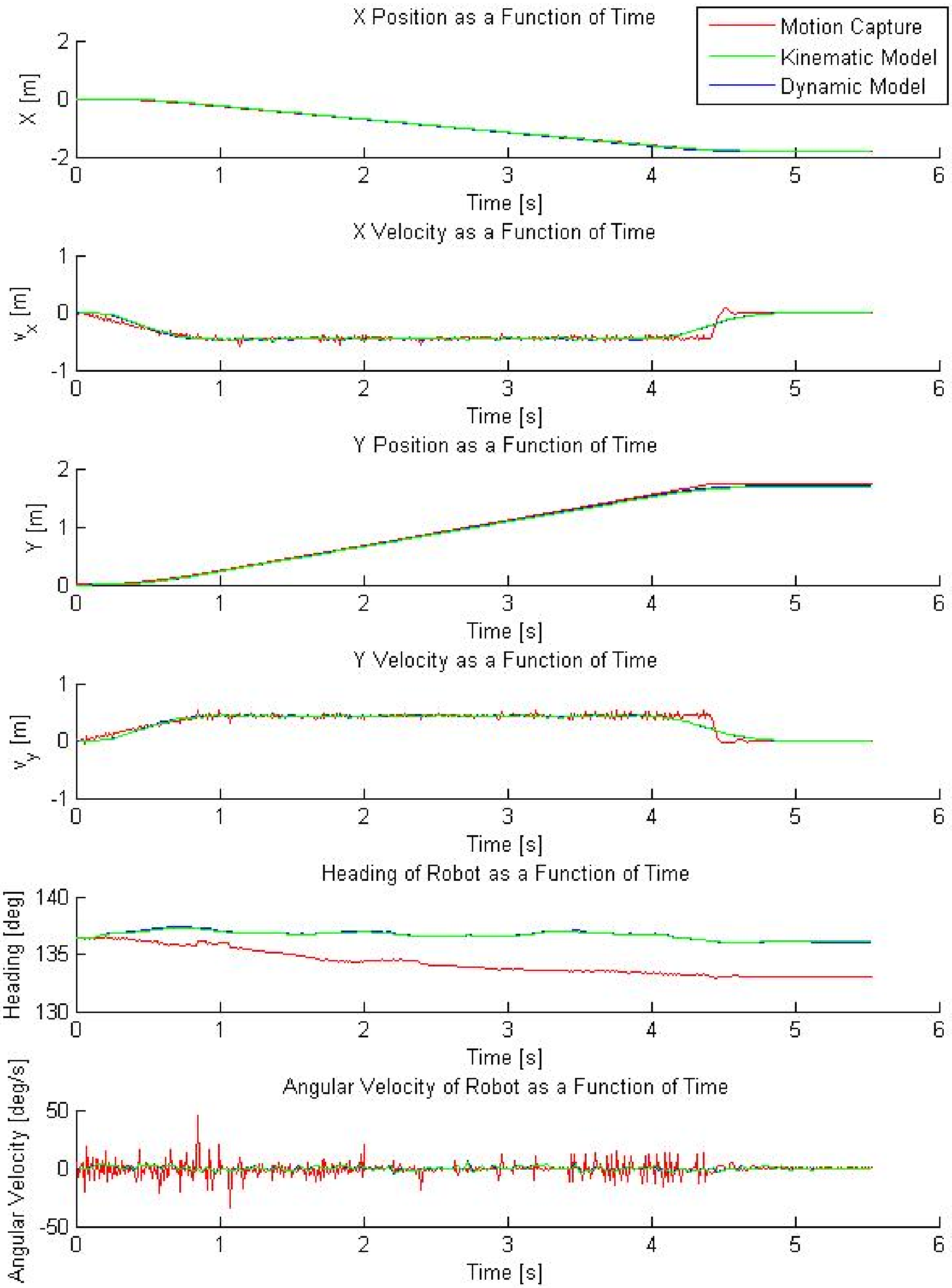


Figure 4.7: The *X* and *Y* position and velocity, robot heading, and angular velocity presented as functions of time for the straight line run.

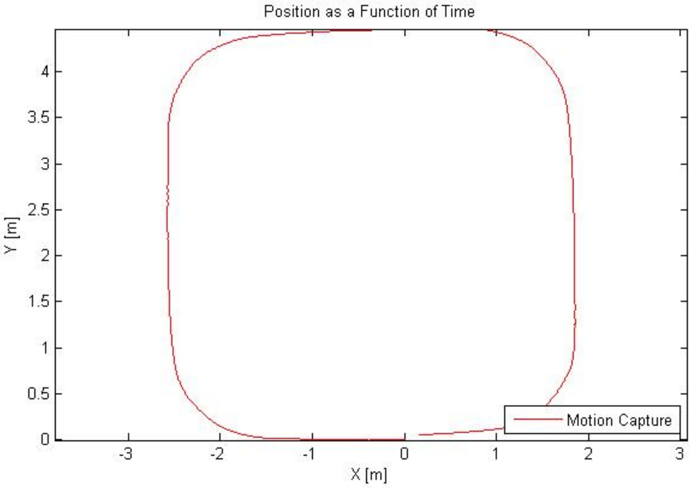


Figure 4.8: The robot's X and Y position as determined by the Vicon motion capture setup are presented as a function of time for the square pattern.

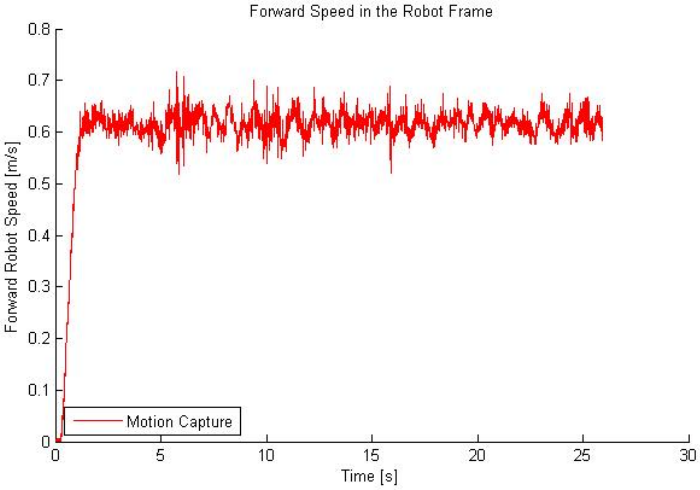


Figure 4.9: The forward speed of the physical platform as determined by the Vicon motion capture setup as a function of time for the square experiment.

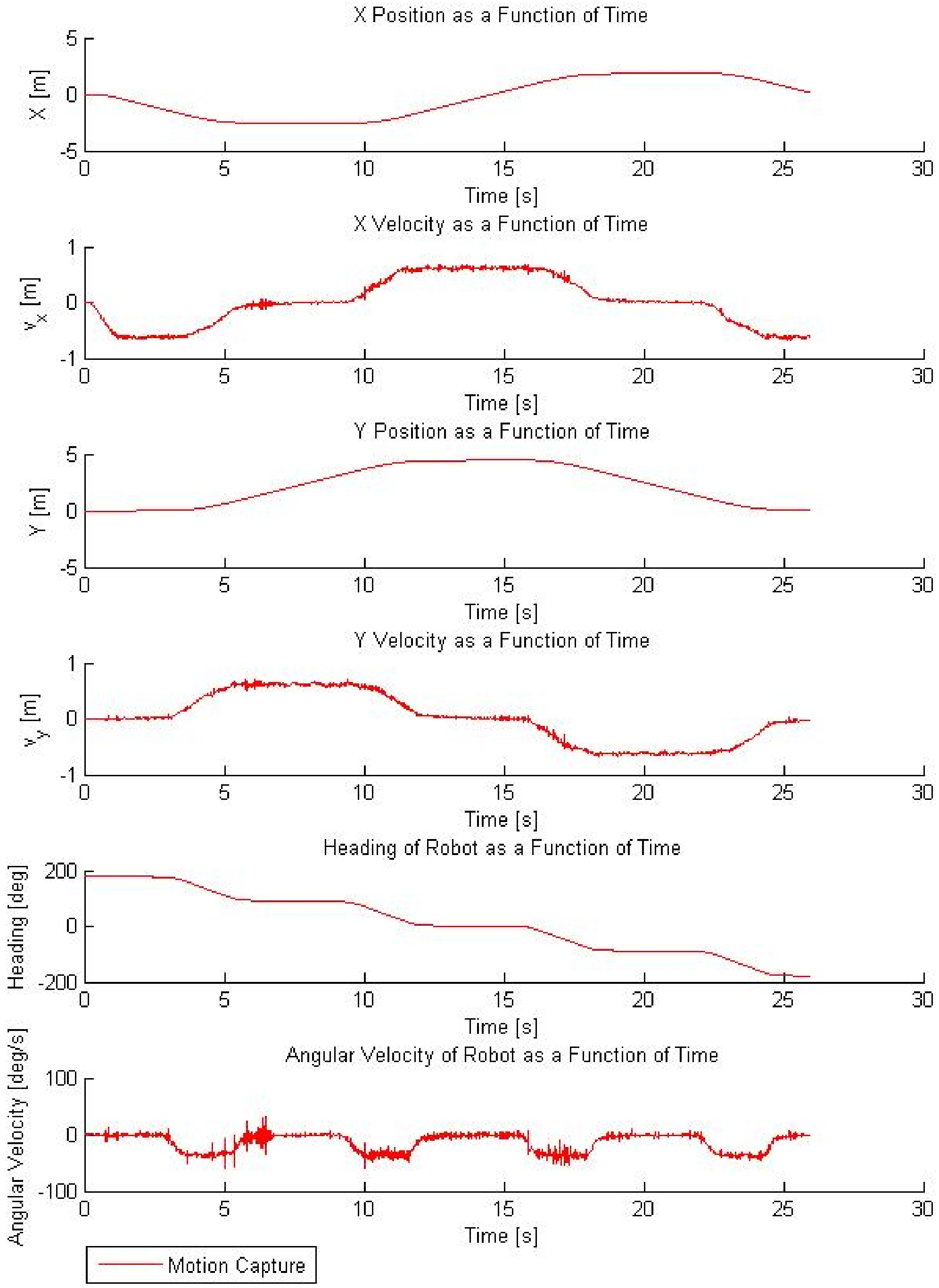


Figure 4.10: The *X* and *Y* position and velocity, robot heading, and angular velocity as determined by the Vicon motion capture setup are presented as functions of time for the square pattern.

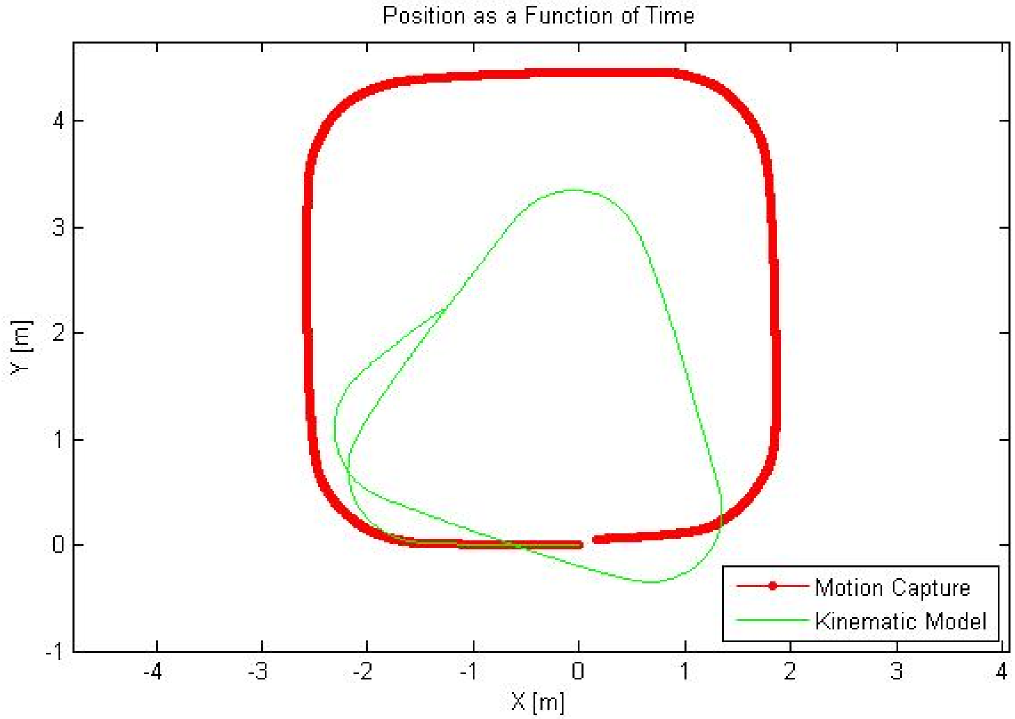


Figure 4.11: Comparison between the motion capture results and the kinematic model results for for the robot's X and Y position are presented as a function of time for the square pattern.

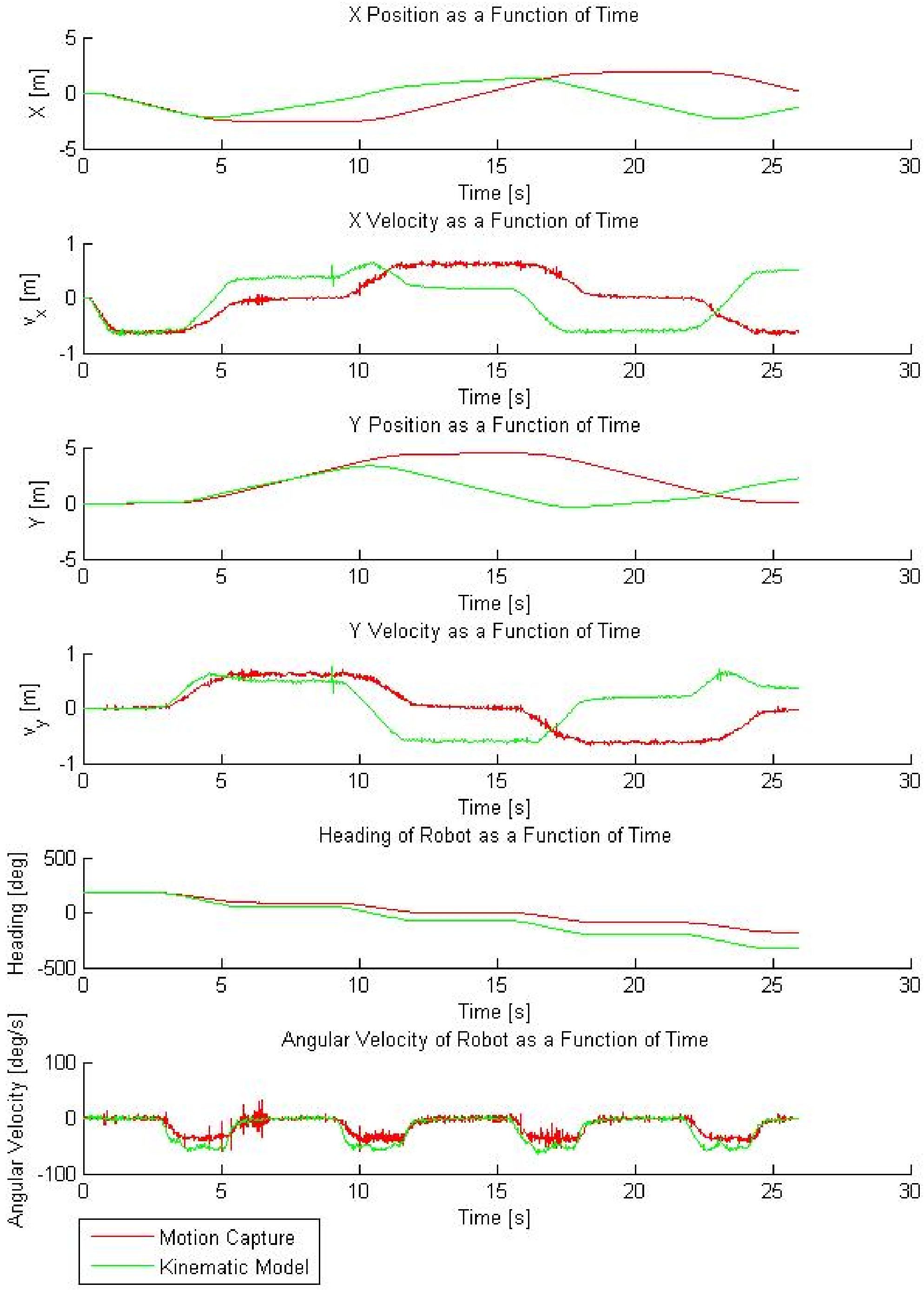


Figure 4.12: Comparison results for motion capture and kinematic model for *X* and *Y* position and velocity, robot heading, and angular velocity as functions of time for the square pattern.

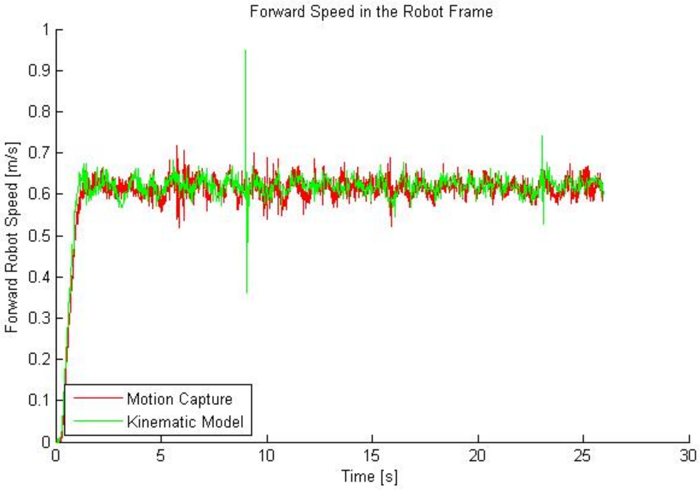


Figure 4.13: Comparison motion capture and kinematic model forward speed as a function of time for the square experiment.

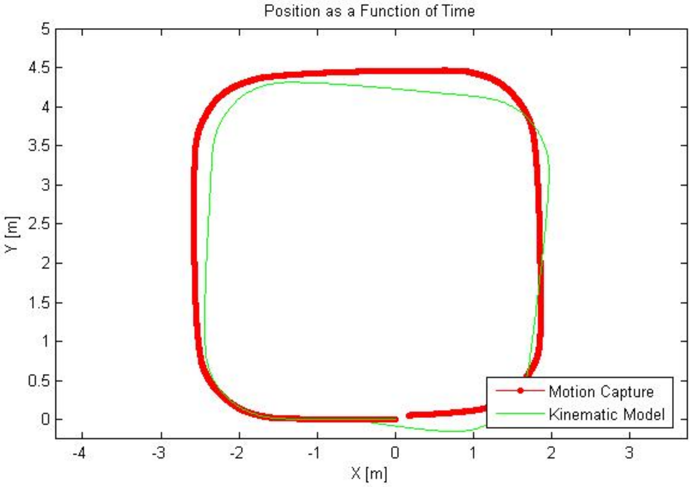


Figure 4.14: Comparison between the motion capture results and the scaled kinematic model results for the robot's X and Y position are presented as a function of time for the square pattern.

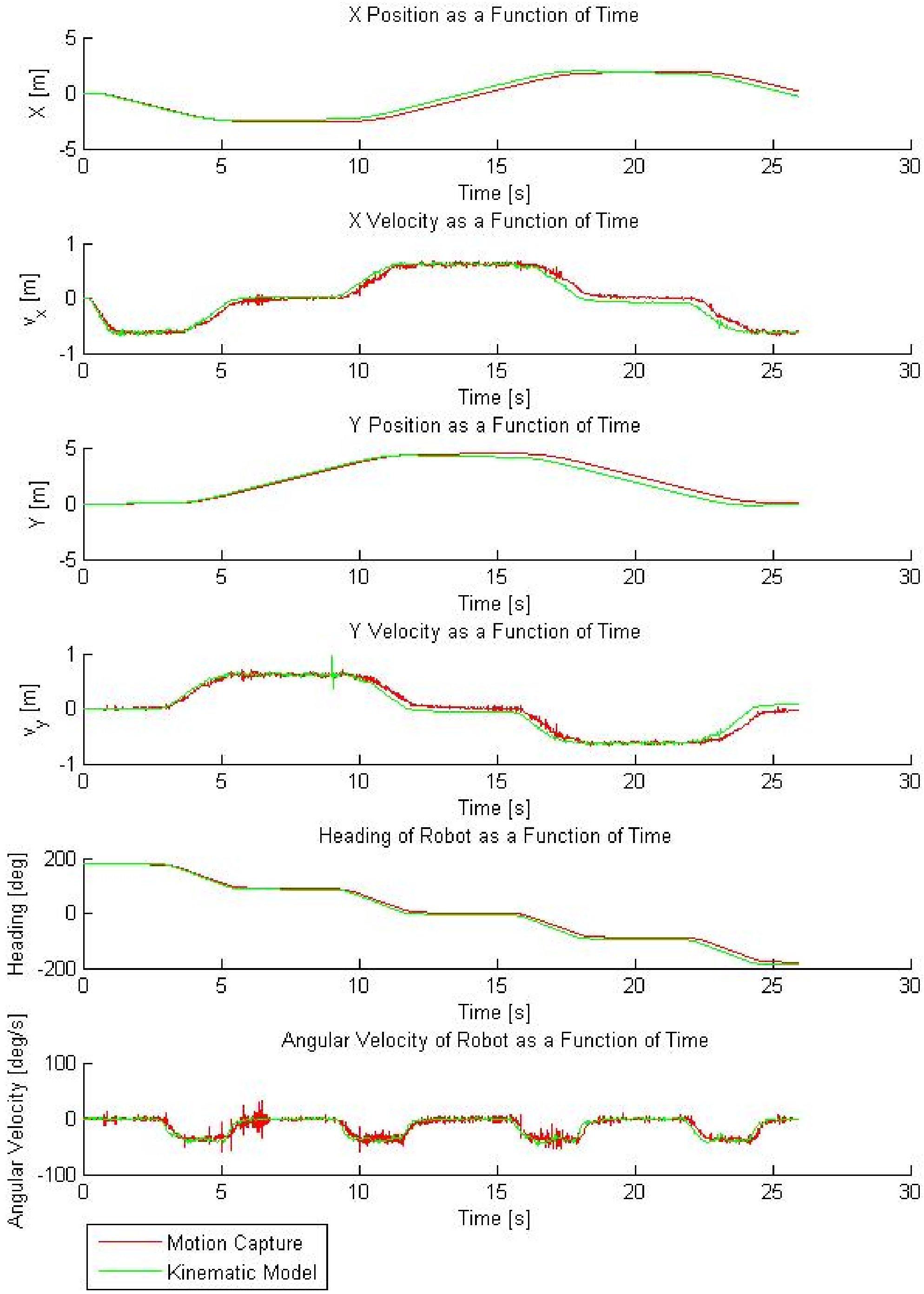


Figure 4.15: Comparison results for motion capture and scaled kinematic model for X and Y position and velocity, robot heading, and angular velocity as functions of time for the square pattern.

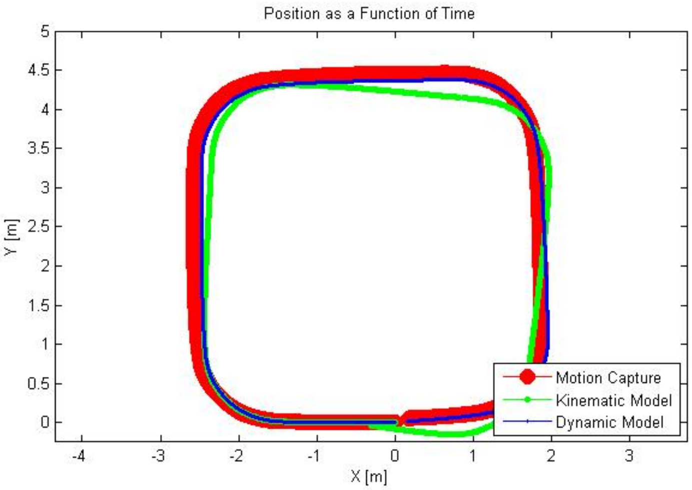


Figure 4.16: Comparison of robot trajectory results for a square run.

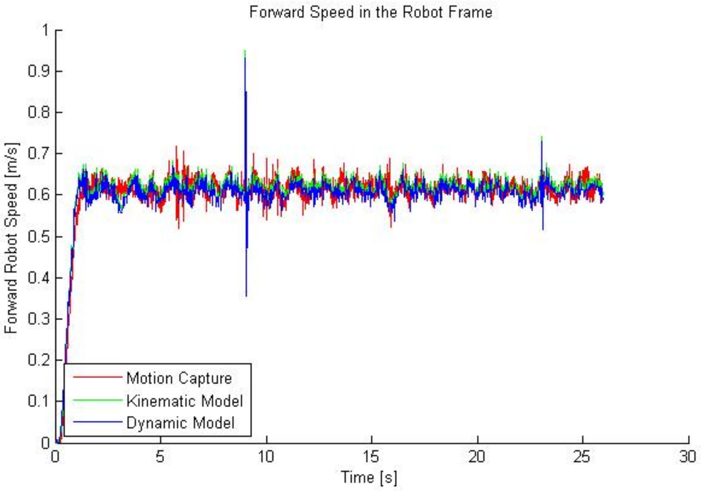


Figure 4.17: Comparison between the motion capture, scaled kinematic model, and scaled dynamic model results for for the robot's X and Y position are presented as a function of time for the square pattern.

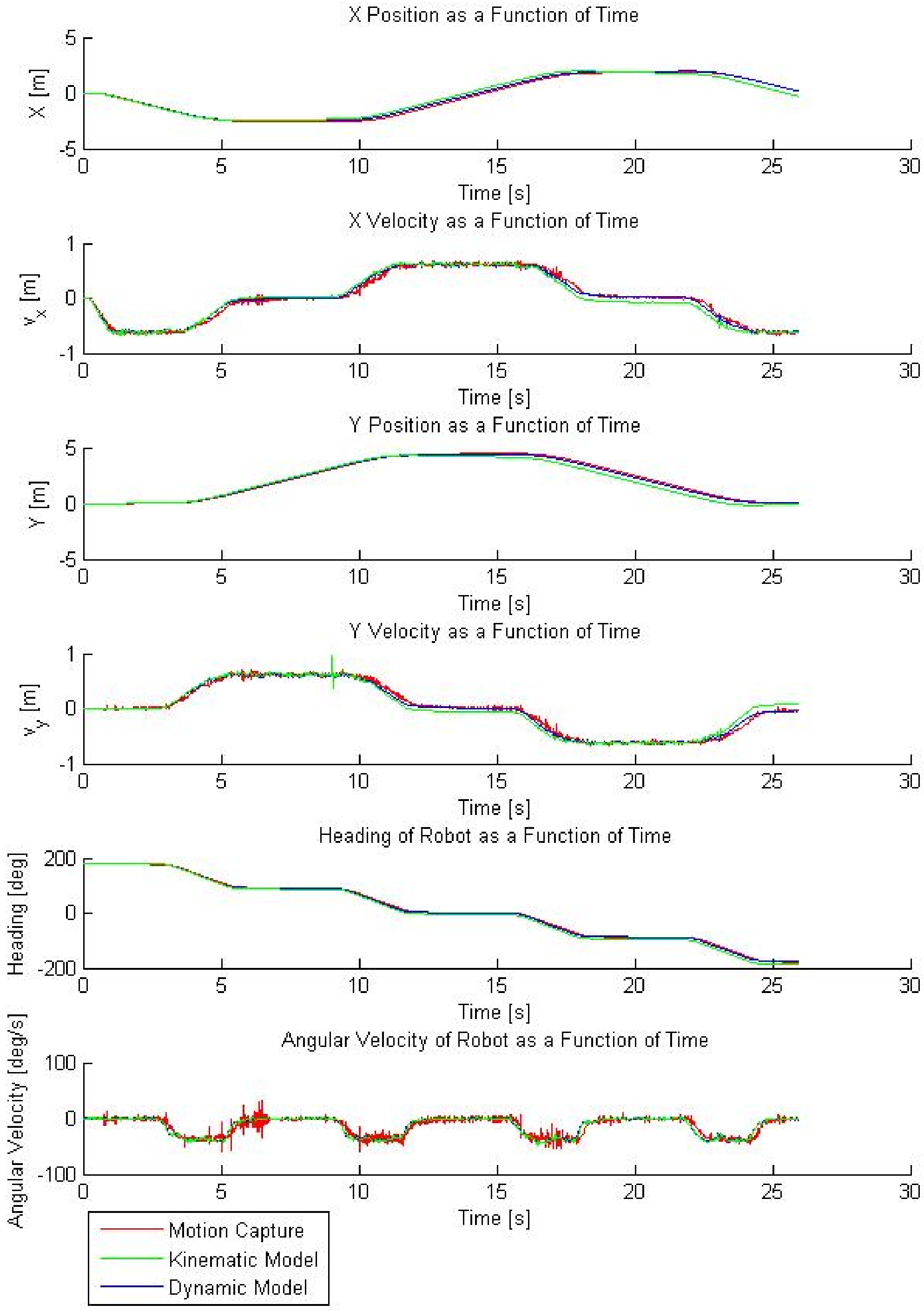


Figure 4.18: Comparison results for motion capture, and scaled kinematic and dynamic models for X and Y position and velocity, robot heading, and angular velocity as functions of time for the square pattern.

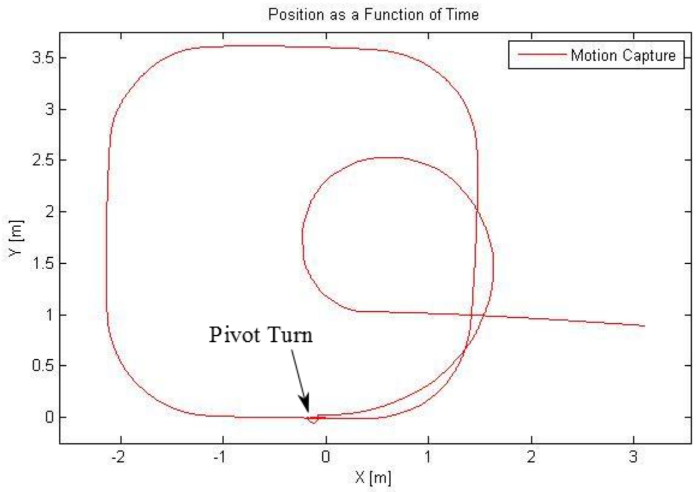


Figure 4.19: Vicon results presented for the complex pattern geometry. Experiment begins at the origin. Note that there is forward motion in the pivot turn observed after the completion of the square.

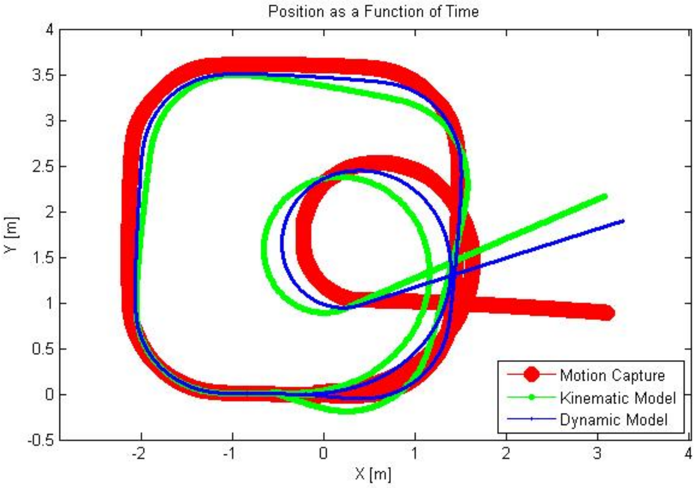


Figure 4.20: Comparison between the motion capture, kinematic model, and dynamic model trajectories. Both models have difficulty with the pivot turn but the dynamic model proves more accurate overall.

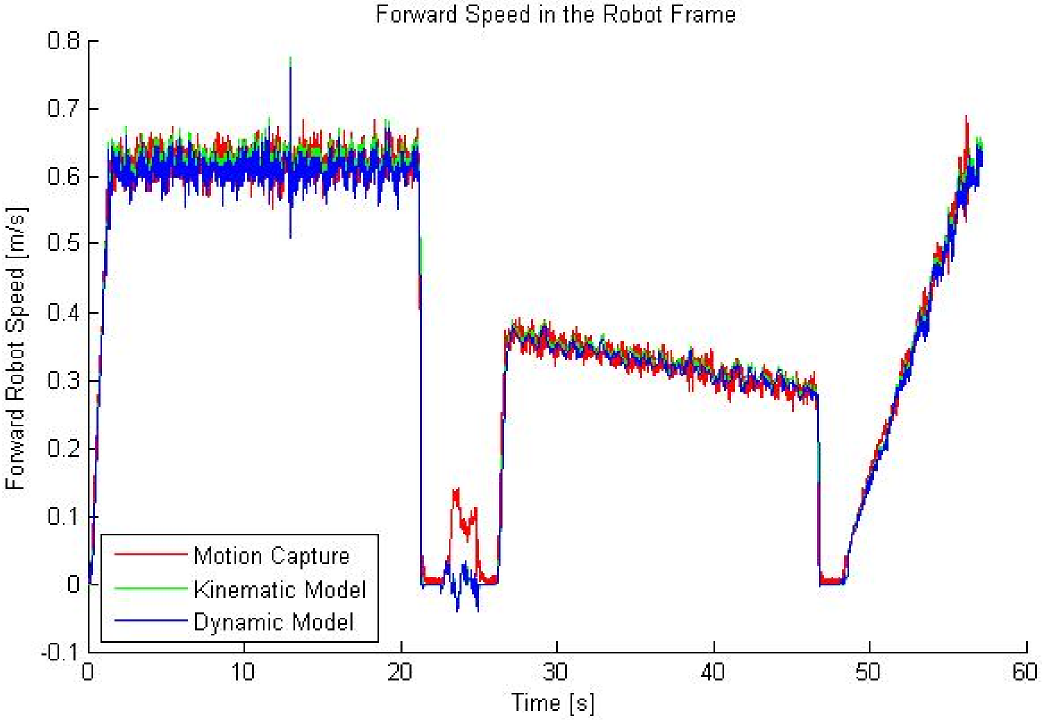


Figure 4.21: The forward speed of the physical platform as a function of time for the square-spiral-line experiment. Results are presented for the motion capture, kinematic, and dynamic models.

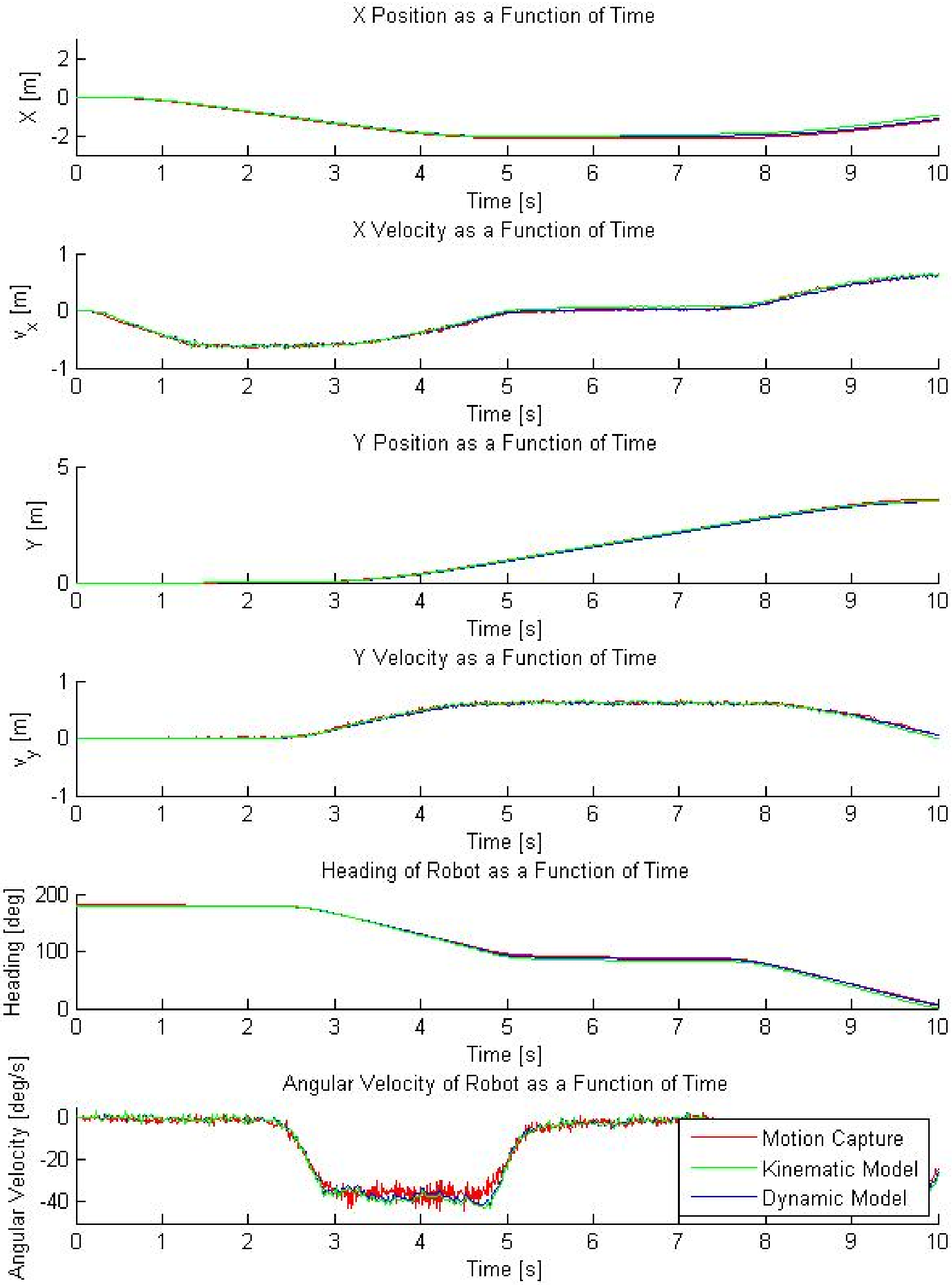


Figure 4.22: Comparison results for motion capture, and scaled kinematic and dynamic models for *X* and *Y* position and velocity, robot heading, and angular velocity as functions of time for the complex pattern. Only the first ten seconds are shown to provide better resolution for the fine grain detail between the various plots.

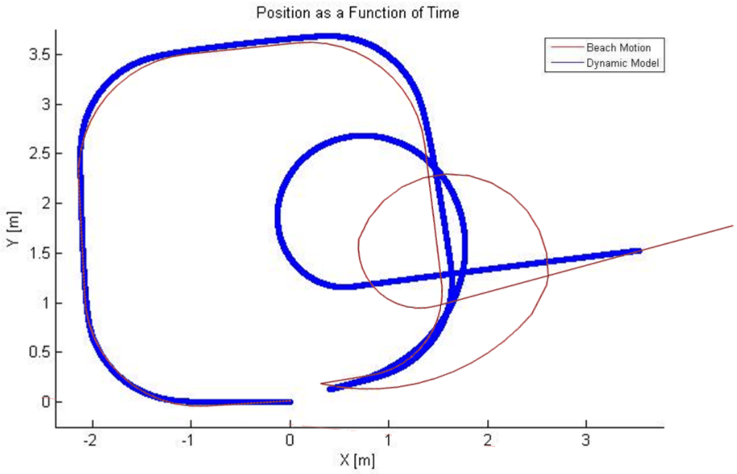


Figure 4.23: The compared tracks of ground truth and the dynamic model prediction for the complex pattern on hard sand. The dynamic model is accurate until the pivot turn in which it over estimates and fails to remain concurrent with the spiral.

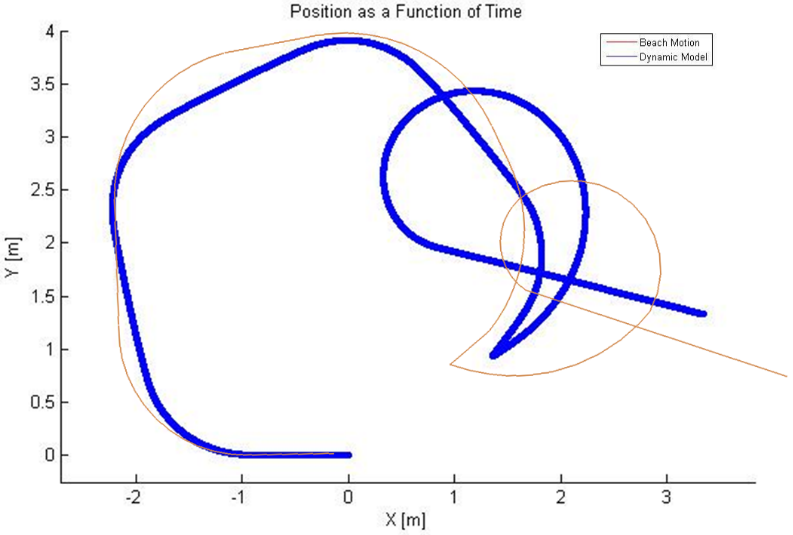


Figure 4.24: The compared tracks of ground truth and the dynamic model prediction for the complex pattern on soft sand. The dynamic model immediately loses accuracy at the first turn and while producing the basic proportions of the pattern, it fails to accurately predict the motion.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusions and Future Work

This chapter provides a discussion of the general successes and deficiencies of the model with specific attention to the recurring differences between motion capture, kinematic and dynamic models. A discussion is also provided of the specific implementation on sand. An extensive amount of lessons learned and critical accomplishments still needed for a final surf zone system is described.

5.1 Discussion of Final Results

As evidenced in Chapter 4, both the kinematic and dynamic models successfully predict and match the physical motion of the vehicle with only minimal sensor input from shaft encoders. Straight line motion on concrete is accurate to within the mechanical asymmetry present in any physical platform. Square patterns with rounded turn corners on concrete are also accurate to within a final position of five centimeters over 12 meters of travel. Both models begin to suffer inaccuracies after extended amounts of turn as shown in the complex pattern. The kinematic model is more susceptible to this error due to its limited ability to handle side slip native to a four wheel skid steered vehicle even with adjustments to its calculation of heading change. The dynamic model is able to better handle this side slip with only minor adjustments to the inertia terms to better match the detailed inertia beyond the simplified value calculated.

During more extensive testing on various types of sand, additional deficiencies are evident with the models. On hard pack sand, the dynamic model shows minor inability to predict the decreased traction and increased side slip during turns. On soft sand, the model breaks down significantly as both traction and side slip become dominant factors. Without reconfiguration of the model, they are unable to account for the significant changes in position over time.

The dynamic model is more effective and useful in that traction and slip are not the only inputs being processed. Based on an applied force, the dynamic model can accommodate external forces such as rotated gravity vectors during slope assents and descents, thrust vectors from waterborne propulsion, additional DOF from non-flat terrain and additional acceleration factors input from onboard sensors such as accelerometers, magnetometers, and gyroscopes.

5.2 Future Areas for Work

Due to the extensive research and exposure to various surf zone robotic systems throughout the stages of this report, a variety of lessons learned and areas for future improvement are obtained. As such, the discussion presents a wide scope of topics beyond the principal topic of modeling. We attempt to summarize the necessary milestones and critical components most important to the successful completion of an operational and deployable surf zone robotic system.

5.2.1 Mathematical Model Improvement and Waterborne Integration

For full implementation onboard a surf zone vehicle, several improvements will be made to the model. The non-flat and sloping terrain of the surf zone requires the full implementation of a six DOF model. Adding these additional DOF will add significant accuracy and prediction as the platform traveres more complex terrain with non-linear traction and slip as well as the freedom of motion in water.

Real time calibration of the model will be capable with feedback from onboard sensors. Motion predicted by the model will be compared to estimated motion from GPS and accelerometers and correction variables will be adjusted to match motion. In addition, an onboard library of established correction variables for known terrain types will be loaded with non-linear relationships to provide discrete values as the vehicles passes through different terrain.

Incorporation of the improved model with the waterborne operation will provide predictive motion through all phases of travel. During periods of shallow submersion as described in the CONOPS, minimal sensor inputs will be available and the ability for the model to provide navigation will be critical.

5.2.2 Realistic Gazebo Simulation of Surf Zone

The Gazebo simulation suite used in ROS is capable of much more advanced terrain modeling and can serve as a capable test bed for sensor implementation, navigation planning and autonomy logic. When fully implemented, complex terrain features and real time sensor feedback will significantly speed up the pace of iteration and avoid the delays and legwork involved with initial physical testing in the surf zone.

A list of critical features of a typical surf zone environment is identified in order to better simulate the details that a vehicle operating would encounter. Though no two coastlines are the same, the following items are identified as critical features for consideration in the simulated environment:

- Gradual and sharply increasing slopes from the waterline
- Small sudden hills and ravines
- Variety of size rocks and protrusions
- Variety of growing and loose vegetation
- Small water run-off paths returning to sea
- Variety of surface material including wet hard pack and soft sand, dirt, rock and shallow water

This list is more exhaustive and beyond the reasonable limitations of Gazebo. Fine grain detail objects such as vegetation and fluids are not within the processing and modeling capabilities of the Gazebo package. Instead, the general topographic and curvature of land will be simulated with a variety of features.

Initial work is accomplished using the Hector Gazebo Package mentioned in Section 3.2.3. A detailed world is spawned with the simulated robot as shown in Figure 5.1. After further refinement and detailing, a challenging representation of a surf zone environment will enable a high performance test arena for future model testing and sensor/autonomy integration.

5.2.3 Autonomy and Navigation

A critical component of the final surf zone robotic system will be a robust and reliable path planning and autonomy suite. The dynamic nature and variety of terrain will pose significant challenges to navigation. To avoid the complete ground up development of this logic, future work will benefit from the use of ROS and its active developer community. Key subsystems from a host of fully developed autonomous control systems can be leveraged into a capable navigation stack tailored for surf zone operations [33]. For example, a complete ROS package can be added to employ the onboard sensors and object detection to build a map of detected terrain in real time that will aid in path planning and navigation as well as being passed to adjacent vehicles for a more complete operational picture of the enviroment [34]. Another benefit that will be gained from the use of ROS for navigation is the direct integration of Gazebo and the simulated surf zone environment mentioned in Section 5.2.2.

Specific modes of operation will be added to address the unique qualities and challenges of the surf zone. Beach terrain tends to transition in uniform areas and directions due to the constant direction of the ocean including its tides, wind and erosion. This also leads to the

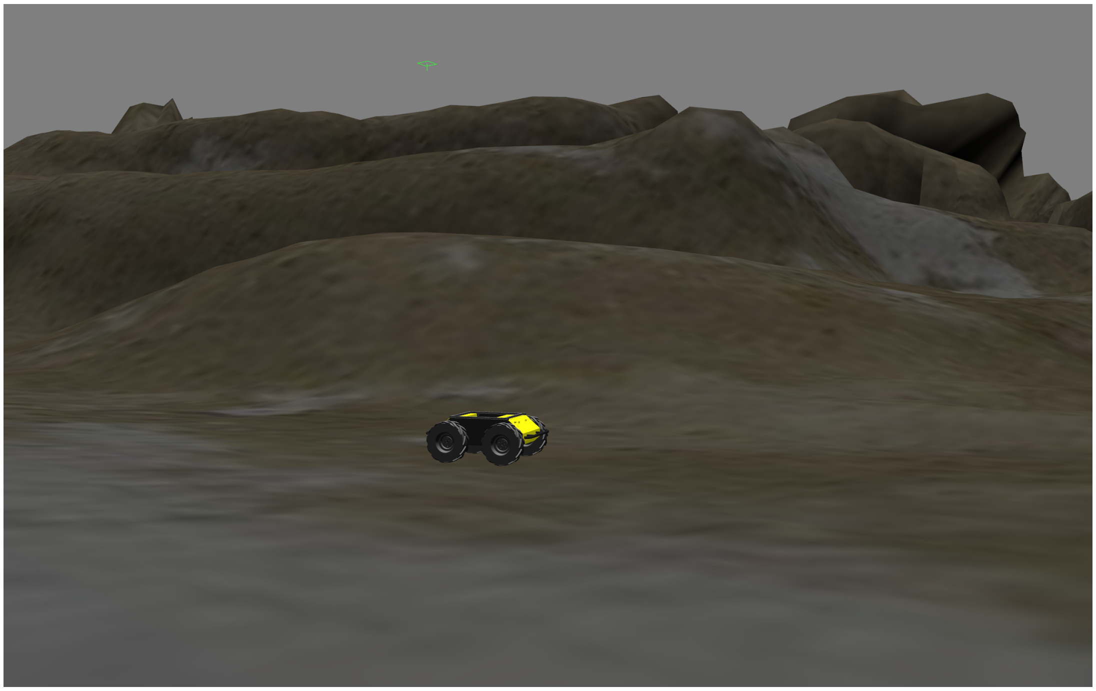
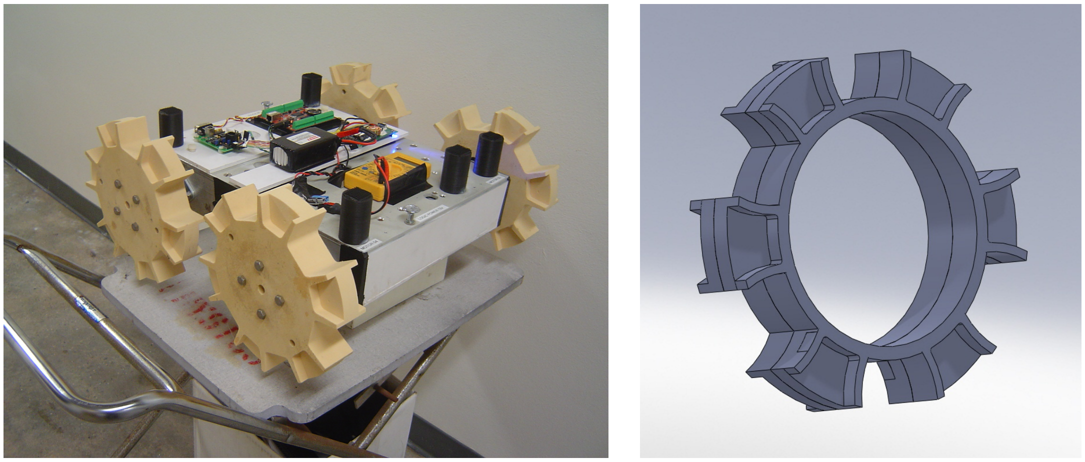


Figure 5.1: Screen shot of a simulated Husky robot in realistic surf zone terrain from the Hector Gazebo package. A very large degree of freedom exist in the specific topography, features and platform that can be combined.

mostly consistently increasing slope and elevation as one moves further away from the surf line. Because of this, it can be expected to traverse sections of terrain with similar qualities for a duration before transitioning to a different terrain type. This quality will allow specific modes of navigation and autonomy to be tuned for the these expected terrains. For example, soft sand will require higher motor velocities to account for the additional slippage but object avoidance can assume that vegetation and trees will not be found in this terrain. Another example is moist hard pack sand found in the intertidal region. This region will afford good traction and very flat geometry but object avoidance will be tuned for known objects such as beached kelp, driftwood, stationary rocks, and the threat of large swells.

5.2.4 Physical Platform Improvement

From the extensive number of platforms researched as well as the various iterations of surf zone vehicles developed at NPS, a cohesive vision of the necessary qualities for a successful platform is gathered.



(a) (b)

Figure 5.2: (a) Test platform with experimental Whegs attached (b) Computer generated design for future Wheg

Simple and Durable

The surf zone environment is an alien and unyielding arena for a robotic system. Adversity such as strong impacts, falls from height, uneven terrain, and exposure to salt water create a multitude of failure points. In order to survive these challenges and remain as a low cost platform envisioned in the CONOPS, a simple and durable design is required. The first design constraint will be the ability to waterproof. From the first iteration, watertight integrity will be maintained and each mechanical or sensor addition will be made watertight before additional features are added. The base shell for the vehicle will be supplied from a commercial waterproof polycarbonate case such as that used by MONTe [12]. The light weight, low cost and durable nature of these cases lends well to the design goals and CONOPS. Suspension systems, while proven capable in platforms like MONTe, expose a myriad of water intrusion spots and points of failure [12]. Instead, a simple passive suspension system will be integrated into the wheel assembly using flexible compliant materials such as rubber or nylon. This will simplify water intrusion to only shaft seals and external sensors. Finally, horizontal symmetry of construction will allow the vehicle to operate in either orientation and improve survivability.

Wheg Optimization and Water Propulsion

The success of Wheg implementation is shown to be successful on a variety of surfaces and scales [5]. A specific analysis of Wheg interaction in the surf zone will provide the necessary qualities and size that will allow the capability to scale the expected terrain and obstacles. The material will provide adequate shock absorption and compliance to flex against obstacles but

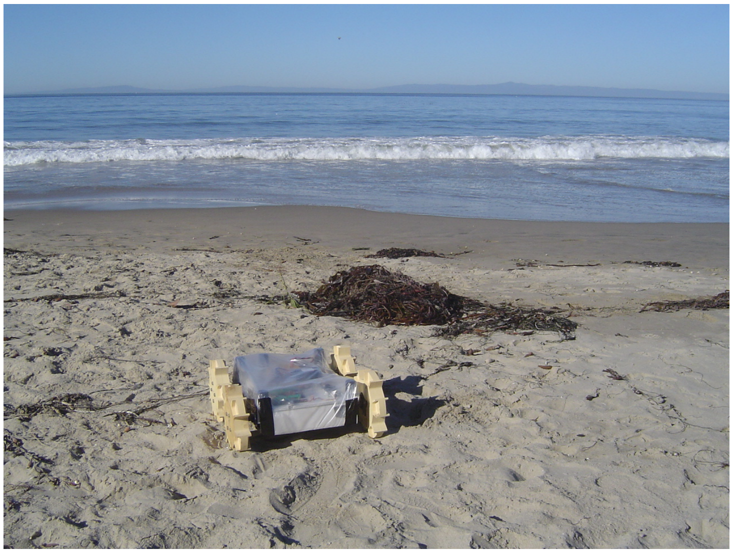


Figure 5.3: Test platform on Del Monte beach using experimental Whegs

maintain shape for adequate traction in loose sand and dirt. The number of spokes, recessed depth, track width and leg shape will all be optimized for the expected terrain, motor power and torque.

Waterborne propulsion design will be accomplished to provide sustained and reliable means of locomotion through swells, breaking waves and floating plant matter. A shrouded jet drive configuration mounted on the rear vertical surface of the main body will provide compact and powerful propulsion while being of minimal intrusion and vulnerability during beach side operations.

Sensor Configuration and Placement

The harsh environment will limit the choice and placement of sensors. Any protrusion outside the safety of the wheel diameter or support structure will be vulnerable to impacts and damage. Any sensor outside the main body will also be either natively waterproof or protected in a watertight housing. With the requirement to sense beyond simple planar detection as required by the non-linearity of the surf zone, a simple, compact and low cost spatial sensor is required. A pivoting LIDAR (Light Detection And Ranging) mounted in a front body configuration will provide horizontally sweeping detection planes with minimal equipment and onboard processing.

References

[1] C. Bernstein, M. Connolly, M. Gavrilash, D. Kucik, and S. Threatt, “Demonstration of surf zone crawlers: Results from AUV Fest 01,” Surf Zone Crawler Group, Naval Surface Warefare Center, Panama City, FL, Tech. Rep. [Online]. Available: http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA479191

[2] R. Quinn, G. Nelson, R. Bachmann, D. Kingsley, J. Offi, and R. Ritzmann, “Insect designs for improved robot mobility,” in *Proceedings of the Climbing and Walking Robots Conference (CLAWAR ’01)*, Karlsruhe, Germany, 2001, pp. 69–76. [Online]. Available: http://biorobots.case.edu/publications/CLAWAR01mobility.pdf

[3] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed., R. Arkin, Ed. Cambridge, Massachusetts: MIT Press, 2004.

[4] R. Quinn, J. Offi, D. Kingsley, and R. Ritzmann, “Improved mobility through abstracted biological principles,” in *IEEE/RSJ International Conference on Intelligent Robots and System*, vol. 3. Switzerland: IEEE, 2002, pp. 2652–2657. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1041670

[5] A. S. Boxerbaum, M. A. Klein, R. Bachmann, R. D. Quinn, R. Harkins, and R. Vaidyanathan, “Design of a semi-autonomous hybrid mobility surf-zone robot,” in *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Singapore: IEEE, Jul. 2009, pp. 974–979. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5229713

[6] Boston Dynamics, “Boston Dynamics RHex,” 2012. [Online]. Available: http://www.bostondynamics.com/robot_rhex.html

[7] Kongsberg Maritime, “REMUS autonomous underwater vehicle,” 2012. [Online]. Available: http://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/D5682F98CBFBC05AC1257497002976E4?OpenDocument

[8] G. Dudek, P. Giguere, C. Prahacs, S. Saunderson, J. Sattar, L.-a. Torres-Mendez, M. Jenkin, A. German, A. Hogue, A. Ripsman, J. Zacher, E. Milios, H. Liu, P. Zhang, M. Buehler, and C. Georgiades, “AQUA: An amphibious autonomous

robot," *Computer*, vol. 40, no. 1, pp. 46–53, Jan. 2007. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4069194

[9] T. Dunbar, "Demonstration of waypoint navigation for a semi-autonomous prototype surf-zone robot," M.S. thesis, Naval Postgraduate School, Jun. 2006. [Online]. Available: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA451347

[10] C. L. Holland, "Characterization of robotic tail orientation as a function of platform position for surf-zone robots," M.S. thesis, Naval Postgraduate School, Jun. 2009. [Online]. Available: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA502235

[11] S. Halle and J. Hickle, "The design and implementation of a semi-autonomous surf-zone robot using advanced sensors and a common robot operating system," M.S. thesis, Naval Postgraduate School, Monterey California, 2011. [Online]. Available: http://www.worldcat.org/title/design-and-implementation-of-a-semi-autonomous-surf-zone-robot-using-advanced\-sensors-and-a-common-robot-operating-system/oclc/743236337

[12] M. Slatt, "Development and testing of a hybrid Wheg platform for autonomous surf-zone operations," M.S. thesis, Naval Postgraduate School, 2011.

[13] J. Rice, B. Creber, C. Fletcher, P. Baxley, K. Rogers, K. McDonald, D. Rees, M. Wolf, S. Merriam, R. Mehio, J. Proakis, K. Scussel, D. Porta, J. Baker, J. Hardiman, and D. Green, "Evolution of Seaweb underwater acoustic networking," in *OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings*, vol. 3. Providence: IEEE, 2000, pp. 2007–2017. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=882235

[14] Willow Garage, "ROS," Menlo Park, Calif.

[15] Willow Garage, "GAZEBO," 2012. [Online]. Available: www.ros.org/wiki/gazebo

[16] Clearpath Robotics, "Clearpath Robotics, Husky A200," 2012. [Online]. Available: http://www.clearpathrobotics.com/husky

[17] U.S. Marine Corps, "Expeditionary Operations," U.S. Government, Washington, D.C., Tech. Rep., 1998. [Online]. Available: http://navsci.berkeley.edu/ma154/19Apr11/MCDP3ExpeditionaryOperations.pdf

[18] J. Yi, H. Wang, J. Zhang, D. Song, S. Jayasuriya, and J. Liu, “Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation,” *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1087–1097, Oct. 2009. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5175357

[19] A. Mandow, J. L. Martinez, J. Morales, J. L. Blanco, A. Garcia-Cerezo, and J. Gonzalez, “Experimental kinematics for wheeled skid-steer mobile robots,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Diego, CA: IEEE, Oct. 2007, pp. 1222–1227. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4399139&contentType=Conference+Publications&searchField%3DSearch_All%26queryText%3Dexperimental+kinematics+for+wheeled+skid+steered

[20] F. Chenavier and J. Crowley, “Position estimation for a mobile robot using vision and odometry,” in *Proceedings 1992 IEEE International Conference on Robotics and Automation*. Nice: IEEE Comput. Soc. Press, 1992, pp. 2588–2593. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=220052

[21] S. B. Niku, *Introduction to Robotics*, 2nd ed., L. Ratts, Ed. Hoboken, New Jersey: John Wiley & Sons, INC., 2011.

[22] W. Yu, “Dynamic modeling and power modeling for robotic skid-steered vehicles,” M.S. thesis, Florida State Univeristy, 2010.

[23] J. R. Taylor, *Classical Mechanics*, 1st ed., L. Young, Ed. Herndon: University Science Books, 2005.

[24] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of Physics*, 9th ed., G. Osnato, S. Johnson, and A. Rentrop, Eds. Jefferson City: John Wiley & Sons, INC., 2011.

[25] The Mathworks INC, *version 7.12.0.635 (R2011a).* Natick, Massachusetts: The MathWorks Inc., 2011.

[26] Q. Ming, “Sliding mode controller design for ABS system,” M.S. thesis, Virginia Polytechnic Institute and State University, Apr. 1997. [Online]. Available: http://scholar.lib.vt.edu/theses/available/etd-5440202339731121/

[27] The Mathworks INC, “Ordinary differential equations.” [Online]. Available: http://www.mathworks.com/help/matlab/math/ordinary-differential-equations.html

[28] M. Quigley, K. Conley, B. Gerky, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: An open-source Robot Operating System," in *International Conference of Robotics and Automation. ser. Open-Source Software workshop*, Kobe, Japan, 2009. [Online]. Available: http://pub1.willowgarage.com/~konolige/cs225B/docs/quigley-icra2009-ros.pdf

[29] Stealth.com INC, "Little PCs," 2012. [Online]. Available: http://stealth.com/littlepc.htm

[30] Canonical, "Ubuntu," London, 2012. [Online]. Available: Ubuntu.org

[31] IRobot Coorperation, "iRobot," 2012. [Online]. Available: http://www.irobot.com/us/robots/Educators/Create.aspx

[32] Vicon, "Motion capture systems from Vicon," 2012. [Online]. Available: www.vicon.com

[33] Willow Garage, "Navigation," 2012. [Online]. Available: www.ros.org/wiki/navigation

[34] Willow Garage, "gmapping," 2012. [Online]. Available: www.ros.org/wiki/gmapping

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Physics Department
   Naval Postgraduate School
   Monterey, California

4. Systems Engineering Department
   Naval Postgraduate School
   Monterey, California

5. Richard Harkins
   Department of Applied Physics
   Naval Postgraduate School
   Monterey, California

6. Timothy Chung
   Department of Systems Engineering
   Naval Postgraduate School
   Monterey, California

7. Steven Martin
   SPAWAR Systems Center, Pacific
   San Diego, California

8. Jeffrey Kline
   CRUSER Director
   Naval Postgraduate School
   Monterey, California